

# Using Curl Tool

## **Objective:**

Learn how to use the Curl tool for manual information gathering.

## **Purpose:**

Curl stands for Client URL. It is a command line tool for getting and sending data including files using URL syntax.

## **Tool:**

Kali Linux

## **Topology:**

We will use Kali Linux for this lab.

## **Walkthrough:**

### **Task 1:**

The general syntax for using curl is the following:

Curl [options] URL

This is a basic syntax that makes the tool quite simple to use. To get some more information on curl and how it is used, type `curl --help` to display the information screen.

Curl can be installed on Linux using the following command:

```
sudo apt-get install curl
```

```
-h, --help <category>      Get help for commands
-i, --include                Include response headers in output
-o, --output <file>         Write to file instead of stdout
-O, --remote-name            Write output to file named as remote file
-s, --silent                Silent mode
-T, --upload-file <file>    Transfer local FILE to destination
-u, --user <user:password>   Server user and password
-A, --user-agent <name>     Send User-Agent <name> to server
-v, --verbose                Make the operation more talkative
-V, --version                Show version number and quit
```

This is not the full help, this menu is stripped into categories.  
Use "--help category" to get an overview of all categories.  
For all options use the manual or "--help all".

```
(olalekan@kali)-[~]
$ sudo apt-get install curl
[sudo] password for olalekan:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.8.0-4).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 858 not upgraded.
```

## Task 2:

The first task we will perform is getting the source code of a site. The first step is to boot your virtual machine and get Kali Linux up and running. Once this is complete, open a terminal and type the following:

curl <https://example.com>

```

$ curl https://example.com
<!doctype html>
<html>
<head>
  <title>Example Domain</title>

  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">
    body {
      background-color: #f0f0f2;
      margin: 0;
      padding: 0;
      font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
    }
    div {
      width: 600px;
      margin: 5em auto;
      padding: 2em;
      background-color: #fdfdff;
      border-radius: 0.5em;
      box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
      color: #38488f;
      text-decoration: none;
    }
  </style>
</head>
<body>
  <div>
    <h1>Example Domain</h1>
    <p>This domain is for use in illustrative examples in documents. You may use this
    domain in literature without prior coordination or asking for permission.</p>
    <pre><code></code></pre>
  </div>
</body>
</html>

```

To save this output to a file, we will use either the “-o” or “-O” option. The lowercase option saves the file with a predefined filename, while the uppercase option saves the file with its original filename. Basically, the lowercase option allows us to specify a file name. This is a useful option if the webpage we are trying to inspect is preventing us from right clicking on the page to view the source code in the browser. Type the following to save your output:

curl -o output.txt <https://example.com>

```

(olalekan@kali)-[~]
$ curl -o output.txt https://example.com
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left     Speed
 100 1256    100 1256    0     0   345      0  0:00:03  0:00:03 --:--:--  346

```

We can see some brief statistic data on this output.

### Task 3:

Curl also provides you with the ability to download multiple files at once. To do this, use multiple -O options, followed by the URL of the file you want to download. For example:

```
curl -O https://arxiv.org/ftp/arxiv/papers/1610/1610.05971.pdf -O
```

```
(olalekan@kali)-[~]
$ curl -O https://arxiv.org/ftp/arxiv/papers/1610/1610.05971.pdf -O https://arxiv.org/pdf/2103.08624.pdf
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100 846k    100 846k    0      0    133k      0  0:00:06  0:00:06 --:--:-- 195k
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100 11083   100 11083    0      0  27193      0  --:--:--  --:--:--  --:--:-- 27230
```

<https://arxiv.org/pdf/2103.08624.pdf>

If your connection drops while downloading a file, you can resume the download with the “-C-” option. This is an especially useful feature when downloading large sized files, ex DVD ISO files, or MP4 video files. This way, if your connection drops when downloading a file, you can resume the download instead of starting from scratch, using, for example:

```
curl -C- -O https://arxiv.org/pdf/2103.08624.pdf
```

### Task 4:

Curl can also be useful for downloading HTTP headers, which is useful when testing a site. To do this,

```
(olalekan@kali)-[~]
$ curl -C- -O https://arxiv.org/pdf/2103.08624.pdf 130

** Resuming transfer from byte position 249
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
0   249    0     0    0     0    0      0  --:--:--  0:00:01  --:--:--    0
curl: (28) Failed to connect to 0.0.0.130 port 80 after 133284 ms: Couldn't connect to server

(olalekan@kali)-[~]
$ curl -C- -O https://arxiv.org/pdf/2103.08624.pdf
** Resuming transfer from byte position 249
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
0   249    0     0    0     0    0      0  --:--:--  0:00:03  --:--:--    0
```

use the following command:

```
(olalekan@kali)-[~]
$ curl -I https://example.com
HTTP/2 200
content-encoding: gzip
accept-ranges: bytes
age: 146691
cache-control: max-age=604800
content-type: text/html; charset=UTF-8
date: Sat, 26 Oct 2024 11:27:13 GMT
etag: "3147526947"
expires: Sat, 02 Nov 2024 11:27:13 GMT
last-modified: Thu, 17 Oct 2019 07:18:26 GMT
server: ECAcc (dcd/7D76)
x-cache: HIT
content-length: 648

(olalekan@kali)-[~]
$
```

`curl -I https://example.com`

This will display many useful pieces of information, such as server info, content type, and content encoding.

### Task 5:

When attempting to download a file or gather other information using curl, you may discover that the target site may be designed to block curl. In this case, it is useful to emulate a browser, such as Firefox, to return the information you are looking for. To do this, use the following command:

```
curl -A "Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0" https://ifconfig.me
```

```
olalekan@kali: ~  
File Actions Edit View Help  
  
(olalekan@kali)-[~]  
$ curl -A "Mozilla/5.0 (x11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"  
https://ifconfig.me  
curl: (1) Protocol "https" not supported  
  
(olalekan@kali)-[~]  
$ curl -A "Mozilla/5.0 (x11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"  
https://ifconfig.me OPTIONS: -Dawt.useSystemAAFontSettings=on -Dwing.aatext=true  
<!DOCTYPE html> Copy files - please wait ... done.  
<html lang="en">  
  
<head>  
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
  <meta http-equiv="content-style-type" content="text/css" />  
  <meta http-equiv="content-script-type" content="text/javascript" />  
  <meta http-equiv="content-language" content="en" />  
  <meta http-equiv="pragma" content="no-cache" />  
  <meta http-equiv="cache-control" content="no-cache" />  
  <meta name="description" content="Get my IP Address" />  
  <meta name="keywords" content="ip address ifconfig ifconfig.me" />  
  <meta name="author" content="" />  
  <link rel="shortcut icon" href="favicon.ico" />  
  <link rel="canonical" href="https://ifconfig.me/" />  
  <title>What Is My IP Address? - ifconfig.me</title>  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <link href="/static/styles/style.css" rel="stylesheet" type="text/css">
```

In this example, remote site <https://ifconfig.me> answers with different messages according to clients' user-agent strings.

## Task 6:

Another important feature of curl is its ability to transfer files. This is useful when interacting with servers through the command line, particularly if you are trying to take advantage of potential vulnerabilities. To access a protected FTP server, use the `-u` option to specify the username and password:

```
curl -u "username:pwd" "ftp://mirrors.sonic.net/knoppix/live.iso"
```

To upload a file to the server, we can use the `-T` option:

```
curl -T file.zip -u "username:password" ftp://mirrors.sonic.net/
```

### Task 7:

Normally, curl denies connection to sites which have invalid SSL certificates. To connect without blocking and getting a warning message, we can use the “-k” option, for example:

```
curl -k http://192.168.1.1/
```

### Task 8:

Curl can also be configured to use a proxy. To do this, use the -x option followed by the proxy URL. For example:

```
curl -x 192.168.0.1:8080 http://example.com/
```

### Task 9:

Curl can also be used for sending HTTP POST data to FORM pages.

In this example, we are sending two parameters, “tfUName” and “tfUPass”, with attached values to “http://testasp.vulnweb.com/Login.asp”.

```
(olalekan@kali)-[~]
$ curl -sk -X "POST" "http://testasp.vulnweb.com/Login.asp" -d "tfUName=admin&tfUPass=none"
<head><title>Object moved</title></head>
<body><h1>Object Moved</h1>This object may be found <a HREF="Default.asp">here</a>
.</body>
```