

# HTTP Requests

In the previous section, we found out that the `secret.js` main function is sending an empty `POST` request to `/serial.php`. In this section, we will attempt to do the same using `cURL` to send a `POST` request to `/serial.php`. To learn more about `cURL` and web requests, you can check out the [Web Requests](#) module.

## cURL

`cURL` is a powerful command-line tool used in Linux distributions, macOS, and even the latest Windows PowerShell versions. We can request any website by simply providing its URL, and we would get it in text-format, as follows:

```
MichaelLuka@htb[/htb]$ curl http://SERVER_IP:PORT/

</html>
<!DOCTYPE html>

<head>
  <title>Secret Serial Generator</title>
  <style>
    *,
    html {
      margin: 0;
      padding: 0;
      border: 0;
...SNIP...
    <h1>Secret Serial Generator</h1>
    <p>This page generates secret serials!</p>
  </div>
</body>

</html>
```

This is the same `HTML` we went through when we checked the source code in the first section.

## POST Request

To send a `POST` request, we should add the `-X POST` flag to our command, and it should send a `POST` request:

```
MichaelLuka@htb[/htb]$ curl -s http://SERVER_IP:PORT/ -X POST
```

Tip: We add the `-s` flag to reduce cluttering the response with unnecessary data

However, `POST` request usually contains `POST` data. To send data, we can use the `-d "param1=sample"` flag and include our data for each parameter, as follows:

```
MichaelLuka@htb[/htb]$ curl -s http://SERVER_IP:PORT/ -X POST -d "param1=sample"
```

Now that we know how to use `cURL` to send basic `POST` requests, in the next section, we will utilize this to replicate what `server.js` is doing to understand its purpose better.

Start Instance

1 / 1 spawns left

Waiting to start...

Questions

Cheat Sheet

Answer the question(s) below to complete this Section and earn cubes!

Target: [Click here to spawn the target system!](#)

+ 1

Try applying what you learned in this section by sending a 'POST' request to '/serial.php'. What is the response you get?

Submit your answer here...

Submit

Hint

Previous

Next

Cheat Sheet

Go to Questions


Table of Contents

Introduction



Introduction	✓
Source Code	✓

Obfuscation


Obfuscation

Code Obfuscation	✓
Basic Obfuscation	✓
Advanced Obfuscation	✓
 Deobfuscation	✓

Deobfuscation Examples


Code Analysis	✓
 <a href="#">HTTP Requests</a>	
 Decoding	

Skills Assessment

 Skills Assessment
Summary

My Workstation

OFFLINE

 Start Instance

1 / 1 spawns left