Curling In The Cloud with localhost

Nous avons face à nous la page contenant le flag. Cependant, un message nous indique que l'accès est refusé

Access Denied, Your IP: 108.161.172.123

Allons à l'index

Enter a URL to fetch

URL: http://csir.dept-info.cros Envoyer

Nous avons encore le même principe que le défi précédent, mais le flag se trouve : /flag.php

Essayons d'y accéder par les adresses de loopback. Le serveur va tenter d'accéder à ses propres ressources. Donc, nous ne devrions pas se faire bloquer par flag.php ??

http://127.0.0.1/flag.php **That IP is a blacklisted cidr (127.0.0.1/24)!**

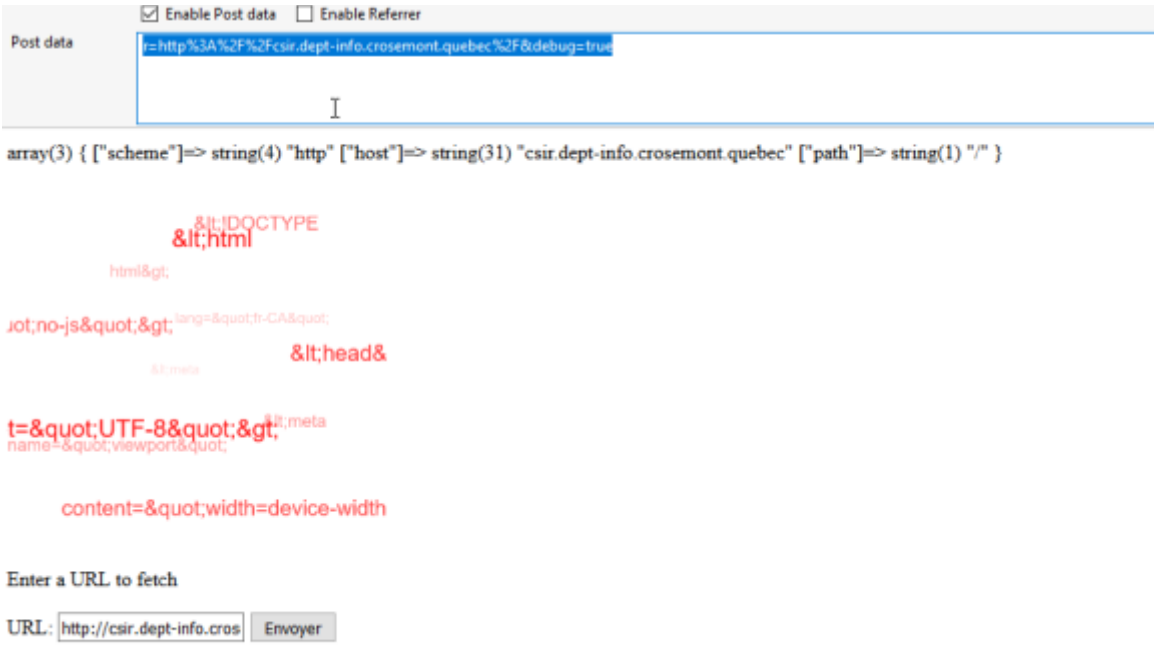
http://169.254.169.254/flag.php **That IP is a blacklisted cidr (169.254.0.0/16)!**

http://0.0.0.0/flag.php **That IP is a blacklisted cidr (0.0.0.0/8)!**

Inspectons le code source !

```
        </div>
    <!---parse_url-->
        <!---debug=false-->
  </body>
'html>
```

Hum essayons de mettre le paramètre debug=true dans notre requête POST.



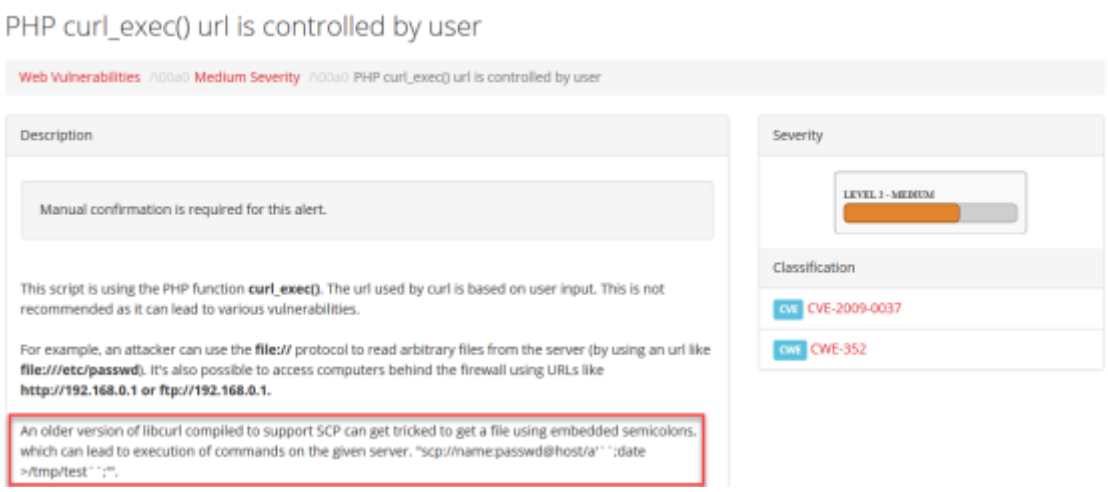
Voilà que le schéma de parse_url nous apparaît ! Encore, faisons une petite recherche sur parse_url et ses vulnérabilités reliées.

Open redirection vulnerability in the Drupal API function drupal_goto (Drupal 6.15 and 5.21)

From: Martin Barbella <martybarbella () gmail com>
Date: Thu, 4 Mar 2010 10:06:07 -0500

```
Open redirection vulnerability in the Drupal API function drupal_goto
(Drupal 6.15 and 5.21)

Discovered by Martin Barbella <martybarbella () gmail com>

Description of Vulnerability:
-----------------------------
Drupal is a free software package that allows an individual or a
community of users to easily publish, manage and organize a wide
variety of content on a website (http://drupal.org/about).

The drupal_goto API function is meant to "send the user to a different
Drupal page. This issues an on-site HTTP redirect. The function makes
sure the redirected URL is formatted correctly"
(http://api.drupal.org/api/function/drupal_goto).

This function will also check $_REQUEST['destination'] and
$_REQUEST['edit']['destination'], and if either of these variables are
set, will override any specified path with the path element of the
associative array returned when passing either request variable
through parse_url.

When a URL such as
"trickparseurl:http://cwe.mitre.org/data/definitions/601.html"; is
passed to PHP's parse_url function, it will return:

array(2) {
  ["scheme"]=>
  string(13) "trickparseurl"
  ["path"]=>
  string(46) "http://cwe.mitre.org/data/definitions/601.html";
}
```

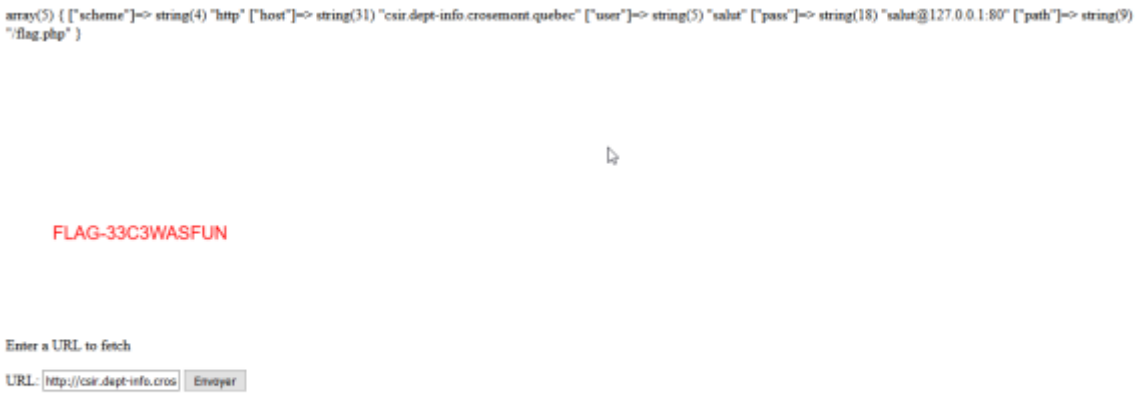
Nous trouvons qu'il est possible de manipuler la syntaxe du schéma construit par parse_url . Regardons la capture d'écran du Writeup précédent.



Essayons de combiner les deux.

r=http://salut:salut@csir.dept-info.crosemont.quebec/flag.php&debug=true

Pouvons-nous ajouter 127.0.0.1 :80 devant @csir.dept-info.crosemont.quebec/



Magie !

Curl interprète notre url de cette façon !

Lors de la validation, l'host sera csir.dept-info.crosemont.quebec.

Cependant, Curl interprétera que l'host est 127.0.0.1 :80 et donc nous il nous permettra d'accéder à flag.php puisque la requête sera faite par lui-même

http://seclists.org/fulldisclosure/2010/Mar/103

https://archive.aachen.ccc.de/33c3ctf.ccc.ac/index.html