UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.
FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN.

Tarea: Natas nivel 11.

Materia: Temas especiales de seguridad informática.

Alumno: Chavez Ortiz Saúl David.

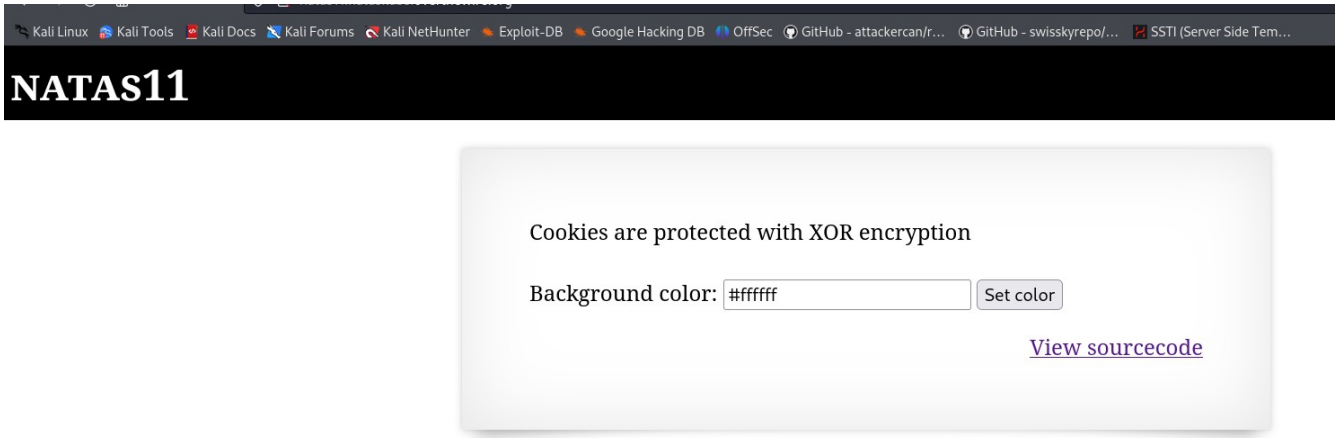
Maestro: Daniel Fernando Palma López.

Carrera: Ingeniería en computación

CDMX 4 de mayo del 2024.

NATAS NIVEL 11.

En este nivel, la página nos indica que las cookies estan cifradas con una encriptación XOR, y en el cual tambien tenemos un formulario con una etiqueta que nos dice “Background color”, tenemos un campo de entrada en donde podemos especificar el color del background y por último tenemos el botón de enviar y establecer. Esto nos da una pista que debemos de trabajar con una cookie.



Si nos vamos al código fuente, podemos encontrar un array asociativo que son los parámetros que se guardan en una cookie, la función del XOR, una función que establece los parametros del array asociativo con la cookie “data”, otra función que establece la cookie “data” pero codificada en JSON, base64 y cifrada en XOR.

Más abajo podemos ver una condicional if, si hay una petición en la variable global $REQUEST, con la clave 'bgcolor', y si es así lo establece en el parametro del array. Por último podemos ver otra condicional if, que si tenemos el parametro 'showpassword' con el valor de "yes", la página nos mostrara la contraseña del natas 12.

```
<script>var wechallinfo = { "level": "natas11", "pass": "<censored>" };</script></head>
<?

$defaultdata = array( "showpassword"=>"no", "bgcolor"=>"#ffffff");

function xor_encrypt($in) {
    $key = '<censored>';
    $text = $in;
    $outText = '';

    // Iterate through each character
    for($i=0;$i<strlen($text);$i++) {
    $outText .= $text[$i] ^ $key[$i % strlen($key)];
    }

    return $outText;
}

function loadData($def) {
    global $_COOKIE;
    $mydata = $def;
    if(array_key_exists("data", $_COOKIE)) {
    $tempdata = json_decode(xor_encrypt(base64_decode($_COOKIE["data"])), true);
    if(is_array($tempdata) && array_key_exists("showpassword", $tempdata) && array_key_exists("bgcolor", $tempdata)) {
        if (preg_match('/^#(?:[a-f\d]{6})$/i', $tempdata['bgcolor'])) {
        $mydata['showpassword'] = $tempdata['showpassword'];
        $mydata['bgcolor'] = $tempdata['bgcolor'];
        }
    }
    }
    return $mydata;
}

function saveData($d) {
    setcookie("data", base64_encode(xor_encrypt(json_encode($d))));
}
```

```
<?
if($data["showpassword"] == "yes") {
    print "The password for natas12 is <censored><br>";
}

?>

<form>
Background color: <input name=bgcolor value="<?=$data['bgcolor']?>">
<input type=submit value="Set color">
</form>
```

Sólucion al nivel de natas 11

Entonces para realizar este nivel de natas 11, como 1er paso necesitamos utilizar el código de la encriptación XOR, y pasarle como argumentos el array asociativo codificado en JSON, y la cookie "data" cifrada que esta alamacenada en la página, importante hay que decodificarla en base 64. Todo esto con el fin de obtener la "$key" del cifrado XOR, el código quedaría así:

```
//Codificamos en formato json el array asociativo de los parametros para pasarla como clave en la función XOR
$defaultdata = json_encode(array("showpassword"=>"no", "bgcolor"=>"#ffffff"));
//Obtenemos la cookie cifrada de la web y la decodificamos de base 64, para pasarla como texto en la función XOR
$cookie = base64_decode("MGw7JCQ5OC04PT8jOSpqdmkgJ25nbCorKCEkIzlscm5oKC4qLSgubjY=");

//Función de encriptación XOR
function xor_encrypt($in, $in2){
$key = $in2;
$text = $in;
```

```
$outText = "";

//Iterate through each character
for($i=0; $i<strlen($text); $i++){
$outText .= $text[$i] ^ $key[$i % strlen($key)];
}

return $outText;
}

//Llamamos a la función XOR y le pasamos la cookie y el arreglo en json, para obtener la clave
$key = xor_encrypt($cookie, $defaultdata);
//Como la clave se repite varias veces, dividimos la cadena larga en partes iguales en un arreglo, y esas partes equivalen a la key
$key = str_split($key, 4);
//Obtenmos la key del arreglo
$key = $key[0];
```

Obtenemos la clave del cifrado y la gurdamos en $key, ahora podemos con ayuda de la $key establecer una cookie cifrada con el parámetro ‘showpassword’ con valor de “yes”, para que la página muestre la contraseña del “natas 12”. Para ello utilizaremos igual la funcion XOR, le pasaremos como argumentos el array asociativo codificado en JSON con el parametro cambiado, la $key, y por ultimo la codificaremos en base 64. El código sería:

```
//Con ayuda de la clave obtenida, modificamos y ciframos los parametros para obtener una cookie cifrada
```

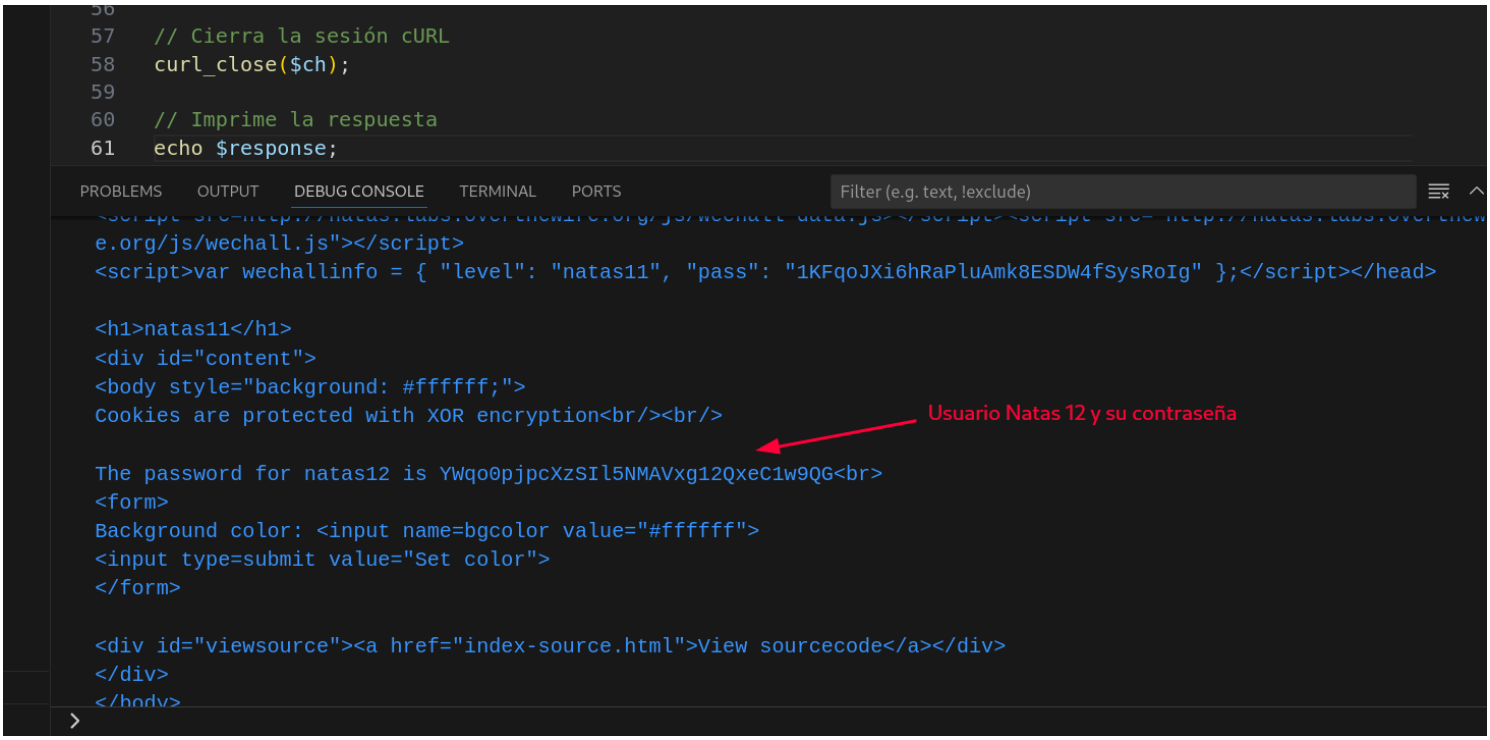
```
echo "Cookie cifrada con el valor 'yes':
".base64_encode(xor_encrypt(json_encode(array("showpassword"=>"yes",
"bgcolor"=>"#ffffff")), $key))."\n";
$pass_cookie =
base64_encode(xor_encrypt(json_encode(array("showpassword"=>"yes",
"bgcolor"=>"#ffffff")), $key));
```

Guardada la cookie cifrada en $pass_cookie, ahora estableceremos esa cookie a la página web con una petición GET. Para poder hacerlo, utilizaremos el módulo o la función CURL de php, además necesitaremos las credenciales del usuario, codificadas en base64 para poder mandarlas con un encabezado de autenticación básica de HTTP, como también en la configuración de CURL estableceremos la cookie cifrada la cual se llama “data”, y otras configuraciones de CURL para la petición GET. El codigo sería:

```
/*PETICIÓN GET A NATAS11*/

//Obteniendo la cookie la pasamos a la página web con una petición GET A
NATAS11
// URL de la página PHP donde se establecerá la cookie
$url = 'http://natas11.natas.labs.overthewire.org/';

//Creedenciales de autorización
$natas11 = "natas11";
$pass_natas11 = "1KFqoJXi6hRaPluAmk8ESDW4fSysRoIg";

//Credenciales coodificadas en base64
$credenciales_base64 = base64_encode("$natas11:$pass_natas11");
```

```
// Inicializa cURL
$ch = curl_init();

// Configura las opciones de cURL
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array("Authorization: Basic ".
$credenciales_base64));
curl_setopt($ch, CURLOPT_COOKIE, "data=".$pass_cookie); // Establece la
cookie
curl_setopt($ch, CURLOPT_HTTPGET, true);

// Realiza la solicitud GET
$response = curl_exec($ch);

// Cierra la sesión cURL
curl_close($ch);

// Imprime la respuesta
echo $response;
```

Una vez hecho la petición, la respuesta se guardara en $response, el cual nos arroja el html de la página con la cookie establecida, en donde este html contiene el password de “natas12”.



Ahora lo que haremos sera filtrar y guardar en una variable solamente la contraseña, para despues hacer otra petición a la página de Natas 12 y logearnos con el usuario y password. El código es:

```
/*Filtra el passNatas12 gracias a la expresión regular (.*)
donde el (.) define cualquier caracter y (*) que se pueden repetir
*/
$patron = '/The password for natas12 is (.*)<br>/';
//La variable $matches es un array que guarda la coincidencia del texto completo y la cadena del pass de natas12 de la expresion regular(.*)
if (preg_match($patron, $response, $matches)) {
// Si se encuentra la cadena, la guardamos en una variable
$pass_natas12 = $matches[1];
echo "Pass de natas12: $pass_natas12";
} else {
echo "No se encontro el pass de natas12";
```

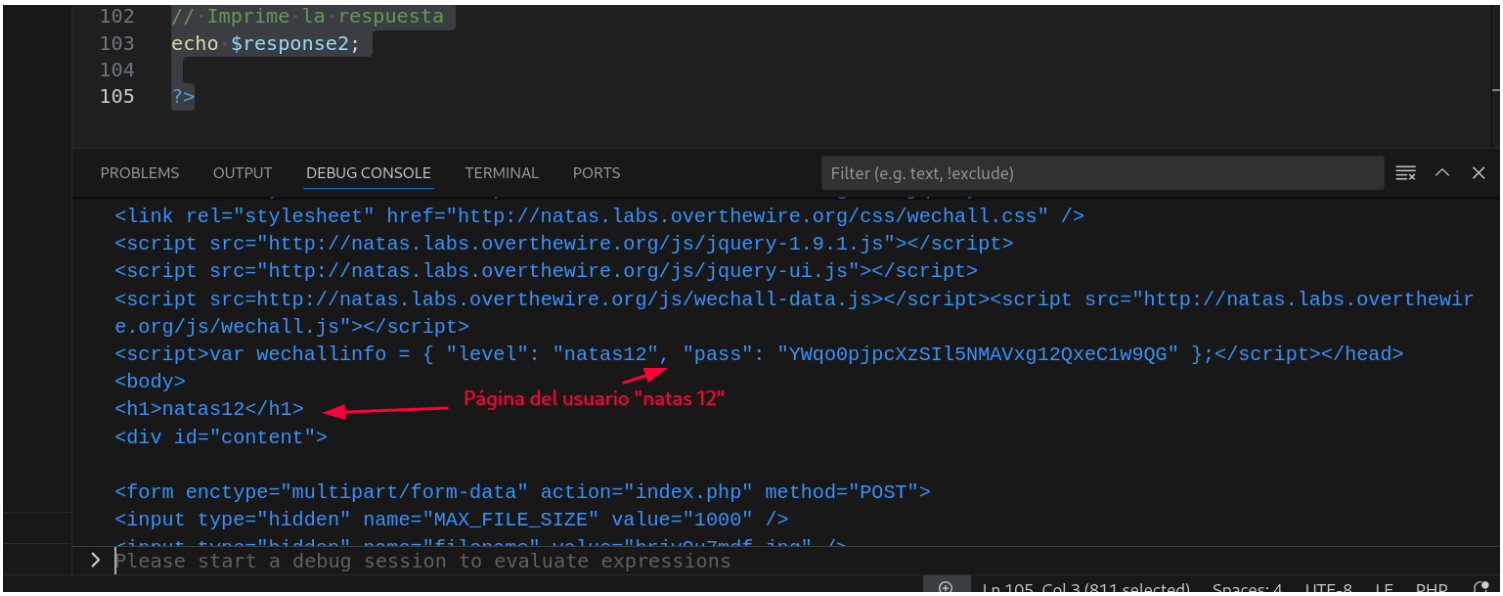
```
}
```

Obtenida la contraseña del natas12 en $pass_natas12, ahora la utilizaremos para hacer una petición GET a su página web y logearnos, para ello utilizaremos el mismo código de la petición anterior, pero en este caso solamente cambiarian las credenciales, que sería las del “natas 12” y su password.

```
/*PETICIÓN GET A NATAS12 PARA LOGEARNOS */
$url2 = 'http://natas12.natas.labs.overthewire.org/';

//Creedenciales de autorización
$natas12 = "natas12";
$pass_natas12 = $matches[1];

//Credenciales coodificadas en base64
$credenciales2_base64 = base64_encode("$natas12:$pass_natas12");

// Inicializa cURL
$ch2 = curl_init();

// Configura las opciones de cURL
curl_setopt($ch2, CURLOPT_URL, $url2);
curl_setopt($ch2, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch2, CURLOPT_HTTPHEADER, array("Authorization: Basic ".
$credenciales2_base64));
curl_setopt($ch2, CURLOPT_COOKIE, "data=".$pass_cookie); // Establece
la cookie
curl_setopt($ch2, CURLOPT_HTTPGET, true);

// Realiza la solicitud GET
$response2 = curl_exec($ch2);
```

```
// Cierra la sesión cURL
curl_close($ch2);

// Imprime la respuesta
echo $response2;

?>
```

Una vez hecha la petición, la respuesta en $response2, nos arroja el html de la página en donde ya estamos logeados como el usuario "natas12"



BIBLIOGRAFÍA.

OverTheWire. (s.f). *Wargames-Natas.* https://overthewire.org/wargames/

OpenAI-ChatGPT. (2023). *Establecer cookie con GET en PHP.* https://chat.openai.com/