RITSEC CTF 2024 Leaked Login Writeup

Manav Malik

Here, we are given the login to this flag service website. We can reasonably assume that we will get the flag once we manage to successfully log into the website. We are able to use the credentials with no issue. From here, we are shown the page where we must enter the OTP to proceed. We see that the OTP is sent to "process.php" to be verified:

```
<form id="otp_form" action="verify/process.php" method="POST"
                    autocomplete="off">
```

Navigating to "process.php," we are met with an "Authenticating..." message which is quickly replaced by "Authentication failed." Checking out the source code here, we see that "process.php" makes a POST request using AJAX to "flag.php" with the data `goodness: "0"`:

```
$(document).ready(function() {
  $.ajax({
    url: 'flag.php',
    type: 'POST',
    data: { goodness: "0" },
    success: function(data) {
      $('#message').text(data);
      $('#message').css('-webkit-user-select', 'text');
      $('#message').css('-ms-user-select', 'text');
      $('#message').css('user-select', 'text');
    },
    error: function(xhr, status, error) {
      if (xhr.status === 401) {
        $('#message').text("Authentication failed.");
      }
    }
  });
});
```

Simply changing this to `goodness: "1"` and using the console to send this request yields an "Invalid token (or timed out)." message, so "flag.php" must have some way of verifying that the request comes from "process.php" in sufficient time.

Instead, we need to intercept the request made by "process.php" and modify it before "flag.php" can review it. To do this, we can load up Burp Suite and open the page in the browser. Once we get to the step where we need to enter the OTP, we will go to the "Proxy" tab and turn on intercept mode to capture all requests. Now, we can enter any six characters and move to the verification process. The first request Burp Suite captures is the one made to "process.php." We have nothing to modify here, so we can just press "Forward" and let it go through. Now, we see the one from "process.php" to "flag.php:"

```
POST /verify/flag.php HTTP/1.1
Host: flag-service.0xmmalik.repl.co
Cookie: PHPSESSID=...
Content-Length: 10
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)...
Sec-Ch-Ua-Platform: "macOS"
Origin: https://flag-service.0xmmalik.repl.co
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://flag-service.0xmmalik.repl.co/verify/process.php
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i
Connection: close

goodness=0
```

We can change the goodness value at the bottom from "0" to "1" and send the request through (hopefully in time for the token to not have expired). At last, the page displays the flag: `RC{z3r0_fact0r_auth3nticati0n}`