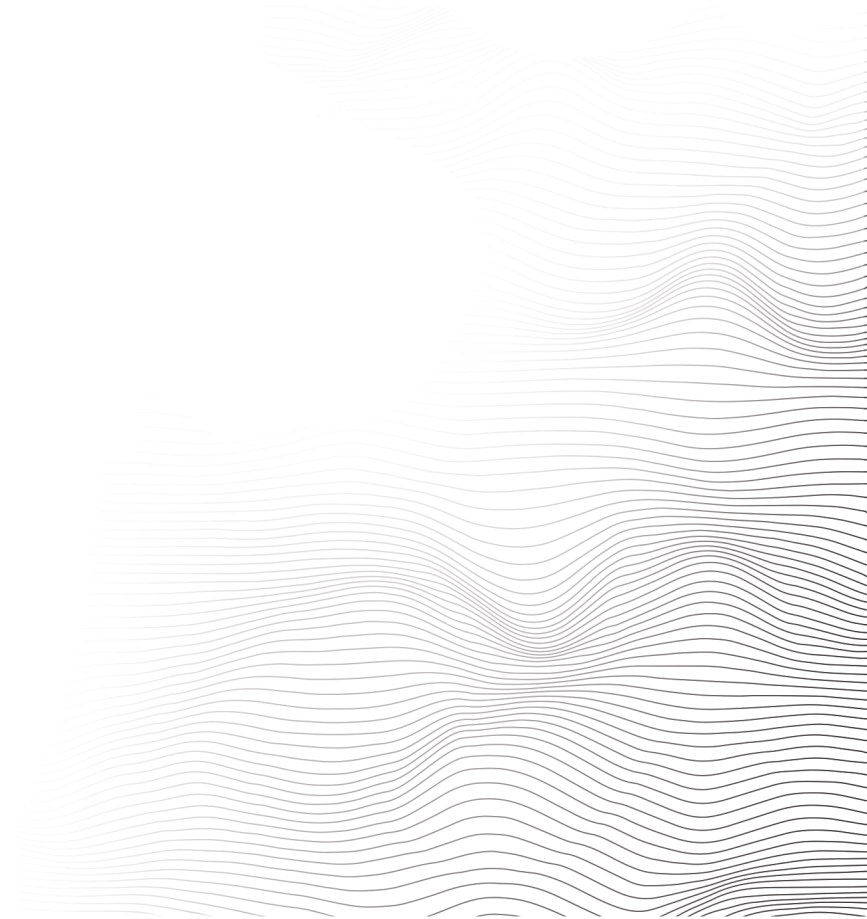
MANDIANT

YOUR CYBERSECURITY ADVANTAGE

Flare-On Challenge 8 Solution

By Nick Harbour

Challenge 1: credchecker

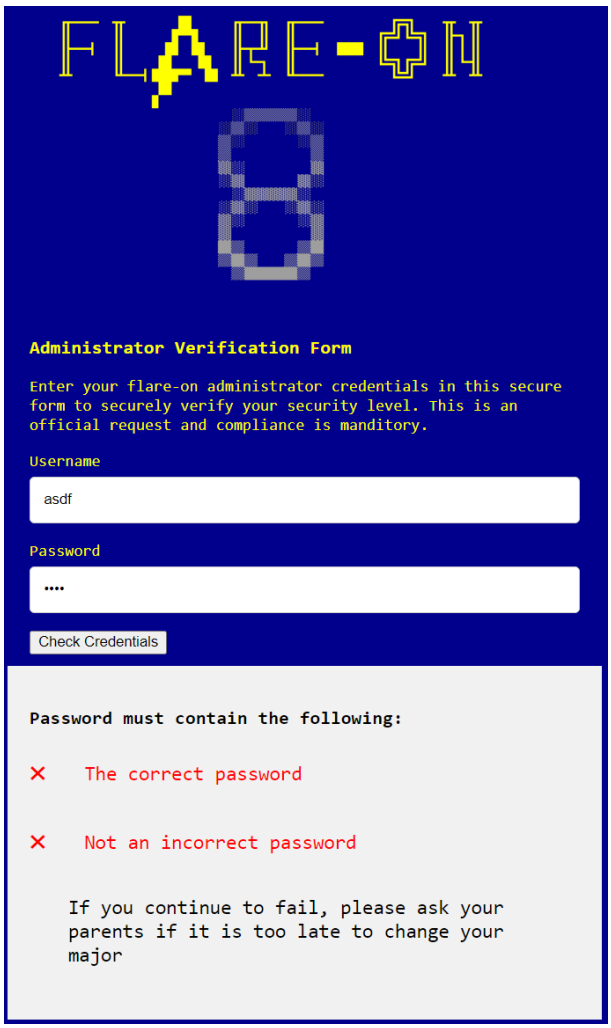


Challenge Prompt

Welcome to Flare-On 8! This challenge serves as your tutorial mission for the epic quest you are about to embark upon. Reverse engineer the JavaScript code to determine the correct username and password the web page is looking for and it will show you the flag. Enter that flag here to advance to the next stage. All flags will be in the format of valid email addresses and all end with "`@flare-on.com`".

Solution

This challenge consists of an HTML file with JavaScript code named `admin.html`, and an `img` directory containing the graphics it uses. Launching the admin.html file with a web browser displays a prompt asking for a username and password. If you enter both it will enable the “Check Credentials” button. If you click this button it determines if you have entered the correct credentials or not. Figure 1 below shows an example of the credential



input screen where incorrect credentials have been entered.

Figure 1: Credential Screen with Failure

To determine what the correct username and password must be to continue, we must reverse engineer the JavaScript code within the `admin.html` file. Line 100 of the file the markup for the "Check Credentials" button is shown below in Figure 2.

```
   <button id="checkbtn" disabled="true" onclick="checkCreds()">Check Credentials</button>
```

Figure 2: Line 100 of admin.html

Line 100 shows that when the "Check Credentials" button is pressed, the JavaScript function named `checkCreds()` will be called. Examining the beginning of the `checkCreds()` function found at line 132 of `admin.html`, shown in Figure 3 below, we see what appears to be a username and password value it is checking for.

```
function checkCreds() {
   if (username.value == "Admin" && atob(password.value) == "goldenticket")
   {
      var key = atob(encoded_key);
      var flag = "";
      for (let i = 0; i < key.length; i++)
      {
      flag += String.fromCharCode(key.charCodeAt(i) ^
                                  password.value.charCodeAt(i % password.value.length))
      }
```

Figure 3: checkCreds() function fragment

If you enter username "`Admin`" and password "`goldenticket`" into the form you will notice that it does not accept the input. While the username `Admin` is compared with a simple string comparison, the password value is compared by first decoding the inputted password with the `atob()` function. `atob()` is a function which decodes Base64 data. Therefore, to pass this check you need to supply a password value that Base64 decodes to "`goldenticket`". Note that it is note acceptable to bypass this check by altering the code as the program decodes the final flag by XORing the encoded flag with the password value you entered. Only the correct password will properly decode the final flag.

The easiest way to determine what data Base64 decodes to "`goldenticket`" is to simply perform the inverse operation, Base64 Encode (or `btoa()` in javascript), on the string "`goldenticket`". Encoding this string results in the Base64 string "`Z29sZGVudGlja2V0`". If you enter this value in the password field and "Admin" in the username field of the form and press the "Check Credentials" button, the code will decode the final flag and advance you to a victory screen where the flag is conveniently displayed, as shown below in .

Figure 4: Victory Screen and Final Flag

©2021 Mandiant Inc. All rights reserved. Mandiant is a registered trademark of Mandiant, Inc. All other brands, products, or service names are or may be trademarks or service marks of their respective owners.