CsecIITB CTF 2020

Category: Web

Challenge: ACM-CSS'20

Points: 500

Description:

> Raj set up his ultra super-duper excruciatingly secure Login system. He even wrote and submitted his paper on it to ACM-CCS'20. The jury consisted of world renowned scientists from the field of Securinti like Akash Dr. Drehan, Dr. Nude Cheetah Nayak, Dr. Lak ssh Kumar and Dr. Him And She Ran Away, sorry, I mean Dr. Himanshu. My bad! I alopogize. Somehow it seems that, even these veritably eminent people couldn't fathom the gravity of this Login system and straightaway rejected it!! I wonder why!? Can you help Raj unearth the cause? The Login system can be found at http://ctf.cseciitb.in:1337

Idea:

On first seeing the page the question appear to has something to do with JS. For web question to inspect the elements is always an good start. In console there is warning about password also. This is a question of client side validation.

On exploring we found that there is a file `svg.js` that is using the password field in some comparisons. For a website to compare the password in JS make them vulnerable to attacks. So mostly what websites do is to send a request and interpret its response.

Let us look closely `svg.js`

```
window.onload = (event) => {
   var f = document.querySelector("form")
   f.addEventListener('submit', function(e){
      e.preventDefault()
      var u = f.elements[0].value
      var oohlala = f.elements[1].value
      var p = ""
      for (var i=0;i<oohlala.length-1;i++){
          p += String.fromCharCode( oohlala.charCodeAt(i)-1 )
       }
       if (u === "imdadmin" && CryptoJS.MD5(p).toString() === "43dd944fa710dbe290fd7b118b7da76e"){
           if (oohlala[oohlala.length-1] === "@"){
               document.location = document.location.href+"/../"+oohlala+".htm"
           }
       }
       else{
           document.getElementById('r').innerHTML = "Wrong Username or
Password!"
18        }
19    })
20 }
```

Start form line:5, here "u" stores the user name and "oohlala" stores the password. The next loop is storing predecessor alphabet of each char in "oohlala" except the last one. If we pass the next if statement we will be redirected to the flag page. From if condition it is clear the "u=imdadmin". The next condition is encrypting "p" with md5 and then comparing to the hex. What we need to do is decrypt the hex and get what p is then every char of "oohlala" will be successor of corresponding index char of p.

Decrypting is not an easy job. But few websites store the string and their encryptions. When a query is made they just return the corresponding string to the hex. On decrypting the hash from here we get "rishabhraj". Which means p="rishabhraj", therfore oohlala="sjtibcisbk". As the last condition says we have to append '@', so finally oohlala="sjtibcisbk@".

Flag

> CSecIITB{Client_S1de_Val1dati0n?Are_y0u_k1dd1ng_me?}