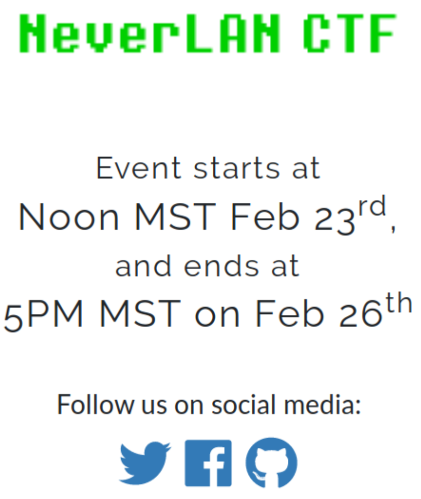
NeverLan CTF 2018 Writeups

Alloysius Goh Follow
Feb 27, 2018 · 3 min read

This is the writeups that my team and I solved for the NeverLan CTF 2018.

The writeups for the recon challenges **will not** be published as it is pure digging for information.
Have fun reading!



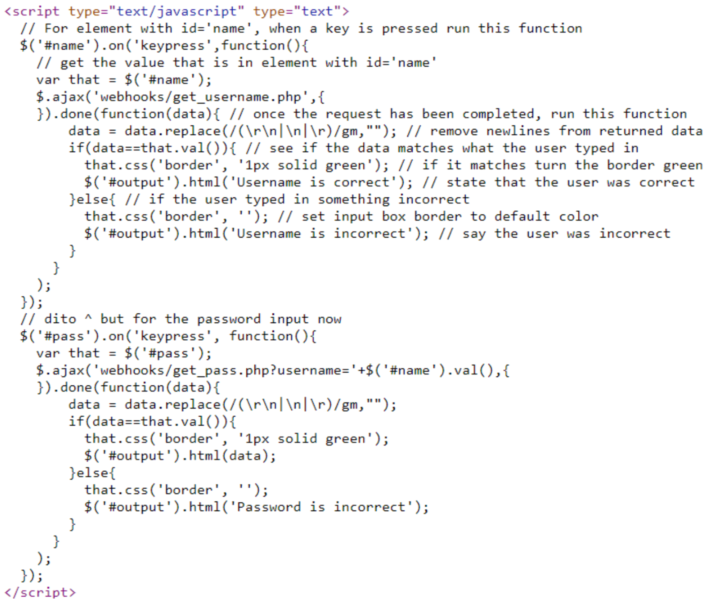
NeverLAN CTF

Event starts at
Noon MST Feb 23rd,
and ends at
5PM MST on Feb 26th

Follow us on social media:

ajax_not_soap—100 Points

First, let’s view the page source.



```
<script type="text/javascript" type="text">
  // For element with id='name', when a key is pressed run this function
  $('#name').on('keypress',function(){
    // get the value that is in element with id='name'
    var that = $('#name');
    $.ajax('webhooks/get_username.php',{
    }).done(function(data){ // once the request has been completed, run this function
        data = data.replace(/(\r\n|\n|\r)/gm,""); // remove newlines from returned data
        if(data==that.val()){ // see if the data matches what the user typed in
          that.css('border', '1px solid green'); // if it matches turn the border green
          $('#output').html('Username is correct'); // state that the user was correct
        }else{ // if the user typed in something incorrect
          that.css('border', ''); // set input box border to default color
          $('#output').html('Username is incorrect'); // say the user was incorrect
        }
      }
    );
  });
  // dito ^ but for the password input now
  $('#pass').on('keypress', function(){
    var that = $('#pass');
    $.ajax('webhooks/get_pass.php?username='+$('#name').val(),{
    }).done(function(data){
        data = data.replace(/(\r\n|\n|\r)/gm,"");
        if(data==that.val()){
          that.css('border', '1px solid green');
          $('#output').html(data);
        }else{
          that.css('border', '');
          $('#output').html('Password is incorrect');
        }
      }
    );
  });
</script>
```

We can see that the username is compared to *webhooks/get_username.php*.
Upon navigation to that site, we find that the username is **MrClean**.

The flag can be found upon navigation to *webhooks/get_pass.php?username=MrClean*.

```
flag{hj38dsjk324nkeasd9}
```

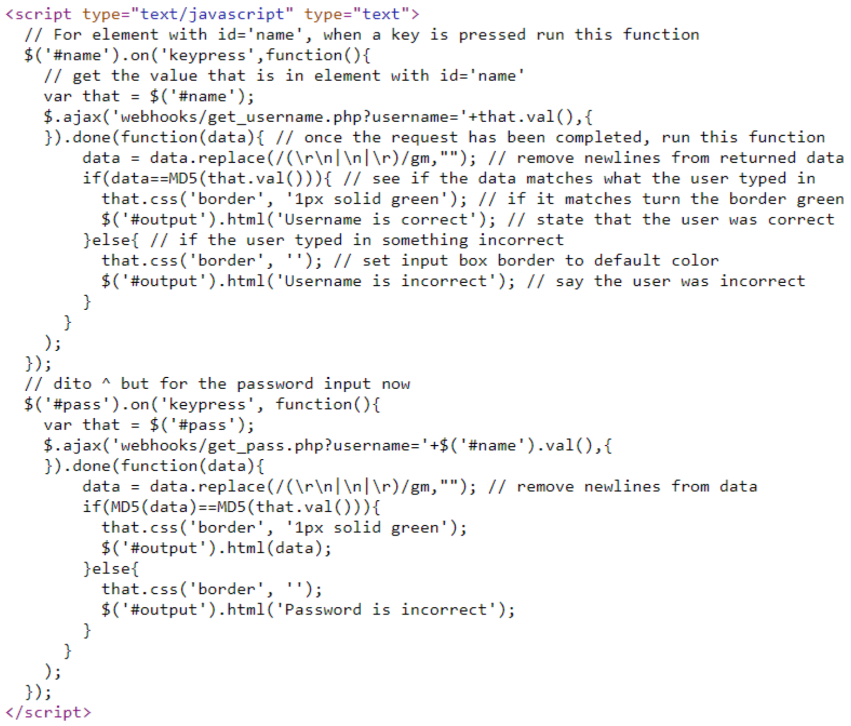
the_red_or_blue_pill—100 Points

When we enter the challenge environment, we can see 2 links : red or blue.
Upon clicking the blue, `?blue` is appended to the back of the url. Since the challenge says we can't select both, I replaced `?blue` to `?blue&red` and the flag appeared.

```
flag{breaking_the_matrix...I_like_it!}
```

ajax_not_borax—200 Points

This challenge is very similar to the other ajax challenge. This just introduces md5 hashing.



```
<script type="text/javascript" type="text">
  // For element with id='name', when a key is pressed run this function
  $('#name').on('keypress',function(){
    // get the value that is in element with id='name'
    var that = $('#name');
    $.ajax('webhooks/get_username.php?username='+that.val(),{
    }).done(function(data){ // once the request has been completed, run this function
        data = data.replace(/(\r\n|\n|\r)/gm,""); // remove newlines from returned data
        if(data==MD5(that.val())){ // see if the data matches what the user typed in
          that.css('border', '1px solid green'); // if it matches turn the border green
          $('#output').html('Username is correct'); // state that the user was correct
        }else{ // if the user typed in something incorrect
          that.css('border', ''); // set input box border to default color
          $('#output').html('Username is incorrect'); // say the user was incorrect
        }
      }
    );
  });
  // dito ^ but for the password input now
  $('#pass').on('keypress', function(){
    var that = $('#pass');
    $.ajax('webhooks/get_pass.php?username='+$('#name').val(),{
    }).done(function(data){
        data = data.replace(/(\r\n|\n|\r)/gm,""); // remove newlines from data
        if(MD5(data)==MD5(that.val())){
          that.css('border', '1px solid green');
          $('#output').html(data);
        }else{
          that.css('border', '');
          $('#output').html('Password is incorrect');
        }
      }
    );
  });
</script>
```

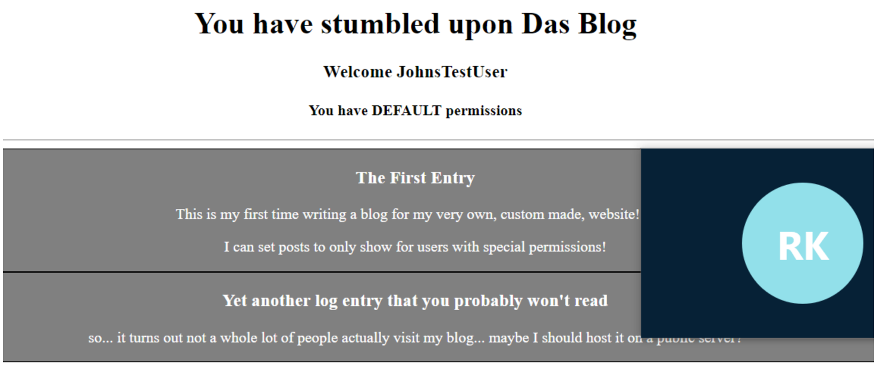
Navigate to *webhooks/get_username.php?username=* again to find the hashed username. We get the MD5 hash of **c5644ca91d1307779ed493c4dedfdcb7.** Crack that hash to reveal the username of **tideade.**

Navigate to *webhooks/get_pass.php?username=tideade* to reaveal the flag encoded with base64. Decode that string to get the flag.

```
flag{sd90J0dnLKJ1ls9HJed}
```

Das_blog — 200 Points

Upon going to the challenge page, we see a login page. View the page source to reveal the credentials to login with. After that, go back to the home screen.



You have stumbled upon Das Blog

Welcome JohnsTestUser

You have DEFAULT permissions

The First Entry

This is my first time writing a blog for my very own, custom made, website!

I can set posts to only show for users with special permissions!

Yet another log entry that you probably won't read

so... it turns out not a whole lot of people actually visit my blog... maybe I should host it on a public server?

My first instinct was to change the cookies. Using the EditThisCookie plugin, change the permission to admin and refresh the page to reveal the flag.

```
flag{C00ki3s_c4n_b33_ch4ng3d_?}
```

cookie_monster — 50 Points

Change the cookie of Red_Guy's_name to Elmo and refresh the page to reveal the flag.

```
flag{C00kies_4r3_the_b3st}
```

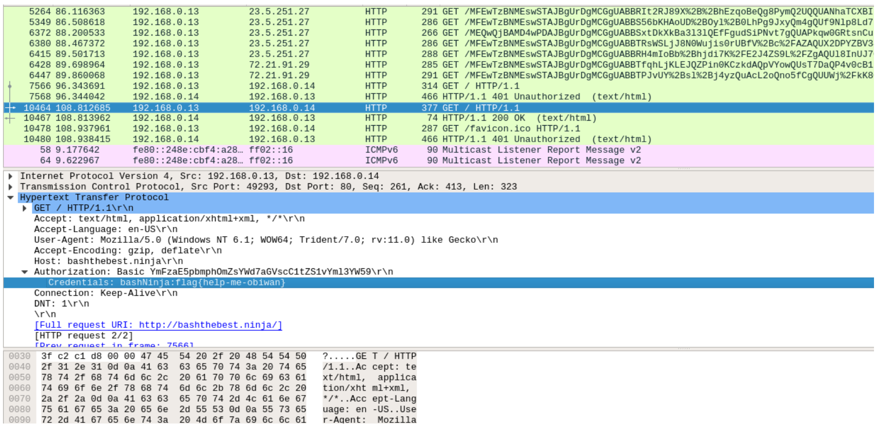
Commitment Issues — 50 Points

Run `strings commitment_issues | grep flag` to get the flag.

```
flag{don't_string_me_along_man!}
```

Encoding != Hashing — 100 Points

Open the file in wireshark and look through the packets with HTTP Protocols.



```
 5264 86.116363   192.168.0.13         23.5.251.27          HTTP     291 GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBRIt2RJ89X%2B%2BhEzqoBeQg8PymQ2UQQUANhaTCXBI
 5349 86.508618   192.168.0.13         23.5.251.27          HTTP     286 GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBS56bKHAoUD%2BOyl%2B0LhPg9JxyQm4gQUf9Nlp8Ld7
 6372 88.200533   192.168.0.13         23.5.251.27          HTTP     266 GET /MEQwQjBAMD4wPDAJBgUrDgMCGgUABBSxtDkXkBa3l3lQEfFgudSiPNvt7gQUAPkqw0GRtsnCu
 6380 88.467372   192.168.0.13         23.5.251.27          HTTP     286 GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBTRsWSLjJ8N0Wujis0rUBfV%2Bc%2FAZAQUX2DPYZBV3
 6415 89.501713   192.168.0.13         23.5.251.27          HTTP     288 GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBRH4mIoBb%2Bhjdi7K%2FE2J4ZS9L%2FZgAQUl8InUJ7
 6428 89.698964   192.168.0.13         72.21.91.29          HTTP     285 GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBTfqhLjKLEJQZPin0KCzkdAQpVYowQUsT7DaQP4v0cB1
 6447 89.860068   192.168.0.13         72.21.91.29          HTTP     291 GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBTPJvUY%2Bsl%2Bj4yzQuAcL2oQno5fCgQUUWj%2FkK8
 7566 96.343691   192.168.0.13         192.168.0.14         HTTP     314 GET / HTTP/1.1
 7568 96.344042   192.168.0.14         192.168.0.13         HTTP     466 HTTP/1.1 401 Unauthorized  (text/html)
10464 108.812685  192.168.0.13         192.168.0.14         HTTP     377 GET / HTTP/1.1
10467 108.813962  192.168.0.14         192.168.0.13         HTTP      74 HTTP/1.1 200 OK  (text/html)
10478 108.937961  192.168.0.13         192.168.0.14         HTTP     287 GET /favicon.ico HTTP/1.1
10480 108.938415  192.168.0.14         192.168.0.13         HTTP     466 HTTP/1.1 401 Unauthorized  (text/html)
   58 9.177642    fe80::248e:cbf4:a28… ff02::16             ICMPv6    90 Multicast Listener Report Message v2
   64 9.622967    fe80::248e:cbf4:a28… ff02::16             ICMPv6    90 Multicast Listener Report Message v2

▸ Internet Protocol Version 4, Src: 192.168.0.13, Dst: 192.168.0.14
▸ Transmission Control Protocol, Src Port: 49293, Dst Port: 80, Seq: 261, Ack: 413, Len: 323
▾ Hypertext Transfer Protocol
  ▸ GET / HTTP/1.1\r\n
    Accept: text/html, application/xhtml+xml, */*\r\n
    Accept-Language: en-US\r\n
    User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko\r\n
    Accept-Encoding: gzip, deflate\r\n
    Host: bashthebest.ninja\r\n
  ▾ Authorization: Basic YmFzaE5pbmphOmZsYWd7aGVscC1tZS1vYml3YW59\r\n
        Credentials: bashNinja:flag{help-me-obiwan}
    Connection: Keep-Alive\r\n
    DNT: 1\r\n
    \r\n
    [Full request URI: http://bashthebest.ninja/]
    [HTTP request 2/2]
    [Prev request in frame: 7566]

0030  3f c2 c1 d8 00 00 47 45  54 20 2f 20 48 54 54 50   ?.....GE T / HTTP
0040  2f 31 2e 31 0d 0a 41 63  63 65 70 74 3a 20 74 65   /1.1..Ac cept: te
0050  78 74 2f 68 74 6d 6c 2c  20 61 70 70 6c 69 63 61   xt/html,  applica
0060  74 69 6f 6e 2f 78 68 74  6d 6c 2b 78 6d 6c 2c 20   tion/xht ml+xml, 
0070  2a 2f 2a 0d 0a 41 63 63  65 70 74 2d 4c 61 6e 67   */*..Acc ept-Lang
0080  75 61 67 65 3a 20 65 6e  2d 55 53 0d 0a 55 73 65   uage: en -US..Use
0090  72 2d 41 67 65 6e 74 3a  20 4d 6f 7a 69 6c 6c 61   r-Agent:  Mozilla
```

Look through the packets and find the flag in the Credentials under Authorization.

`flag{help-me-obiwan}`

Zip Attack—100 Points

This challenge requires a tool called **pkcrack.** This tool allows you to decrypt the other files in a zip file as long as you have a copy of an encrypted file and its unencrypted version.

Run `./pkcrack -C [path to encrypted zip file] -c supersecretstuff/sw-iphone-first-order.jpg -P [path to unencrypted zip file] -p sw-iphone-first-order.jpg` to get the three keys needed to decrypt the rest of the contents.

Run `./zipdecrypt <key0> <key1> <key2> [path to encrypted zip] [path to enencrypted zip]`
Now, unzip the unecrypted zip to reveal flag.txt. Open the text file to reveal the flag.

`flag{plaintext-attacks-are-cool!}`

even more basic math with some junk—100 Points

This challenge requires programming knowledge and we used python to solve this challenge. We implemented regex to find all the number and add them together.
`34659711530484678082` is the answer.

The script can be found at
 https://github.com/alloygoh/NeverLanCTF-

2018/tree/master/even%20more%20basic%20math%20with%20some%20junk