

1 / 6

```

c\x8c\x73\x45\x32\x5b\x8c\x2a\xf1\x2f\x3f\x57\x6e\x04\x3d\x16\x75\x67\x16\x4f\x6d\x
x1c\x6e\x40\x01\x36\x93\x59\x33\x56\x04\x3e\x7b\x3a\x70\x50\x16\x04\x3d\x18\x73\x3
7\xac\x24\xe1\x56\x62\x5b\x8c\x2a\xf1\x45\x7f\x86\x07\x3e\x63\x47'

function _ (x, y) {
  return x ^ y
}
function __ (y) {
  var z = 0
  for (var i = 0; i < y; i++) {
    z += Math.pow(2, i)
  }
  return z
}

function ___ (y) {
  var z = 0
  for (var i = 8 - y; i < 8; i++) {
    z += Math.pow(2, i)
  }
  return z
}
function ____ (x, y) {
  y = y % 8
  i = __ (y)
  i = (x & i) << (8 - y)
  return i + (x >> y)
}
function _____ (x, y) {
  y = y % 8
  i = ___ (y)
  i = (x & i) >> (8 - y)
  return (i + (x << y)) & 0x00ff
}
function _____ (x, y) {
  return _____ (x, y)
}
function _____ (_____, key) {
  _____ = ''
  _____2 = ''
  for (var i = 0; i < _____.length; i++) {
    c = _____.charAt(i)
    if (i != 0) {
      t = _____.charAt(i - 1) % 2
      switch (t) {
        case 0:
          cr = _ (c, key.charAt(i % key.length))
          break
        case 1:
          cr = ____ (c, key.charAt(i % key.length))
          break
      }
    }
  }
}

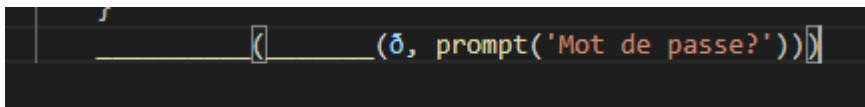
```

```

    } else {
      cr = _(c, key.charCodeAt(i % key.length))
    }
    _____ += String.fromCharCode(cr)
  }
  return _____
}
function _____ (p) {
  var η = 0
  for (var i = 0; i < p.length; i++) {
    η += p['charCodeAt'](i)
  }
  if (η == 8932) {
    var ϕ = window.open(
      '',
      '',
      '\x77\x69\x64\x74\x68\x3d\x33\x30\x30\x2c\x68\x65\x69\x67\x68\x74\x3d\x32\x20\x30'
    )
    ϕ.document.write(p)
  } else {
    alert('Mauvais mot de passe!')
  }
}
_____(_____(δ, prompt('Mot de passe?'))))
````

```

Có rất nhiều hàm được định nghĩa và cuối cùng có lời gọi hàm:



Hai hàm được gọi là:

```

58 function _____ (p) {
59 var η = 0
60 for (var i = 0; i < p.length; i++) {
61 η += p['charCodeAt'](i)
62 }
63 if (η == 8932) {
64 var ϕ = window.open(
65 '',
66 '',
67 '\x77\x69\x64\x74\x68\x3d\x33\x30\x30\x2c\x68\x65
68)
69 ϕ.document.write(p)
70 } else {
71 alert('Mauvais mot de passe!')
72 }
73 }

```

```

36 function _____ (_____, key) {
37 _____ = ''
38 _____2 = ''
39 for (var i = 0; i < _____.length; i++) {
40 c = _____.charCodeAt(i)
41 if (i != 0) {
42 t = _____.charCodeAt(i - 1) % 2
43 switch (t) {
44 case 0:
45 cr = _(c, key.charCodeAt(i % key.length))
46 break
47 case 1:
48 cr = _____(c, key.charCodeAt(i % key.length))
49 break
50 }
51 } else {
52 cr = _(c, key.charCodeAt(i % key.length))
53 }
54 _____ += String.fromCharCode(cr)
55 }
56 return _____
57 }

```

Ban đầu hàm \_\_\_\_\_() được gọi truyền vào hai tham số là `đ` và `prompt('Mot de passe?')` - input người dùng nhập vào

Đọc sơ hàm ta thấy hàm duyệt qua tất cả kí tự của `đ` và nếu đó là kí tự đầu tiên thì sẽ được XOR với kí tự tương ứng chuỗi input người dùng nhập vào (`key[i]`), qua hàm `fromCharCode` và cộng dồn vào \_\_\_\_\_ (dòng 54); ngược lại nếu không là kí tự đầu thì sẽ dựa vào tính chẵn lẻ của kí tự index liền trước \_\_\_\_\_ lấy `CharCode`, nếu chẵn sẽ tiến hành phép XOR đơn giản 2 kí tự tương ứng `đ` và `input`, nếu lẻ sẽ gọi tới hàm \_\_\_\_\_ thực hiện 1 loạt tính toán dài dòng khác.

Kết quả tiếp tục được truyền vào hàm \_\_\_\_\_, duyệt qua cả chuỗi truyền vào và tính tổng `CharCode` các kí tự, nếu bằng 8932 thì hiện thông báo thành công, ngược lại hiện thông báo thất bại.

Liên tục liệt kê các key giá trị, tham khảo hint sử dụng bằng `blasting` có được 2 key giá trị thỏa mãn tổng kiểm tra đã phân tích phía trên bằng 8932 là: `+3` và `yasmin`

Tiến hành nhập thử:

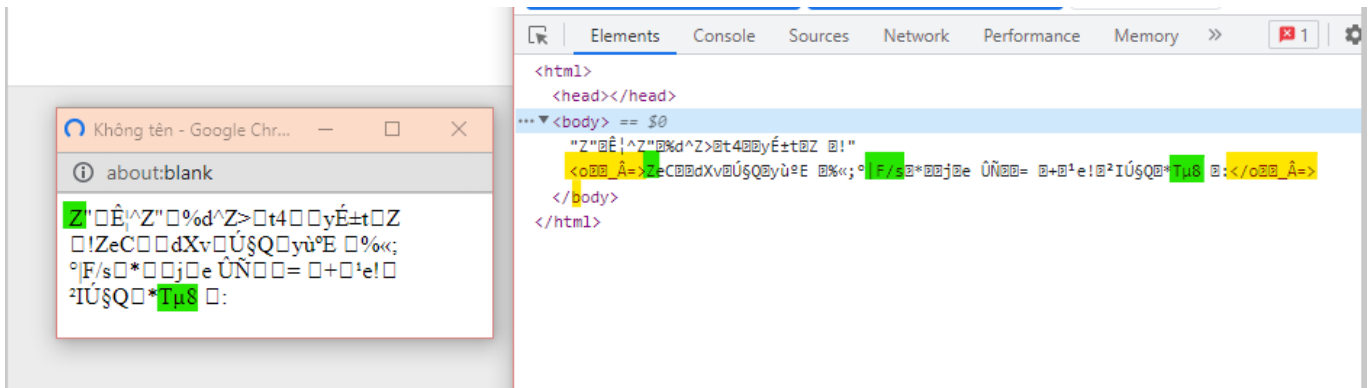
The screenshot shows a web browser window with the address bar displaying 'web-client/ch17/ch17.html'. Below the browser window, a modal dialog box is open. The dialog has a title bar that says 'challenge01.root-me.org cho biết'. Inside the dialog, the text 'Mot de passe?' is displayed above a text input field. The input field contains the text '+3'. At the bottom of the dialog, there are two buttons: 'OK' and 'Hủy'.

Dựa vào hint ta cần enable popups

*NB : You will have to enable popups in order to solve this challenge!*

Start the challenge

Sau đó một popups hiện lên. Thử inspect ta thấy có vẻ hai khối màu vàng là một cặp thẻ đóng mở, bên trong chứa nội dung hiển thị lên popups. Dự đoán hai cặp thẻ đó có thể là "<html></html>"



Nhìn lại luồng challenge thì ta có thể dự đoán challenge yêu cầu nhập pass, gọi hàm

\_\_\_\_\_ (\_\_\_\_\_, prompt('Mot de passe?')) thực hiện giải mã ð với key là pass nhập vào, sau đó in ra plaintext nếu tổng kí tự ascii plaintext = 8932. Tuy nhiên có nhiều pass thỏa mãn nhưng sẽ chỉ có một pass cho ra plaintext có thể đọc được (có nghĩa).

Trở lại phân tích popups, Ở đây ta thử trường hợp đơn giản là các ký tự của pass (key) khi XOR với ð phải là chẵn để hàm XOR được áp dụng trên cả chuỗi. Để ý khi thực hiện XOR là XOR ð[i] với charcode key[i % key.length], trường hợp key ngắn hơn thì sẽ quay lại XOR từ đầu key.

```
case 0:
 cr = _(c, key.charCodeAt(i % key.length))
 break;
```

Do vậy để tìm key thử lấy 6 kí tự đầu suy đoán từ plaintext <html> XOR với 6 ký tự đầu của ð ta được "MyP4sS".


Dùng password submit challenge. Thành công !

## Javascript - Obfuscation 4

50 Points 

Author

aaSSfxxx, 18 July 2011

Level 



### Statement

Find the password.

*NB : You will have to enable popups in order to solve this chall*

[Start the challenge](#)

### Validation

Well done, you won 50 Points

Don't forget to give your opinion on the challenge by voting :-)



twittez le !

Enter password

.....

**Flag:** MyP4sS