# University of Calgary Webmail Bug Report

Yang Li and Michael E. Locasto

Dec 1, 2011

**Program Name**: U of C Webmail System, SquirrelMail
**Version**: Unknown
**Software Source**: http://www.ucalgary.ca/it/email/webmail

**Problem Description**:

I am graduate student from Computer Science Department in University of Calgary. I have been suffering a web email problem after I become a U of C student. I have a user ID “li4” for my email account. Every time, when I click “add” to attach a file to the current email, there will pop out a warning window, see Figure I. Also, there is no any draft saved in my account. If I click “OK” in this window, then every field in “To:”, “Cc:”, “Bcc:”, “Subject:”, and the content field will be filled with “undefined”. For sure, I lose everything I wrote.

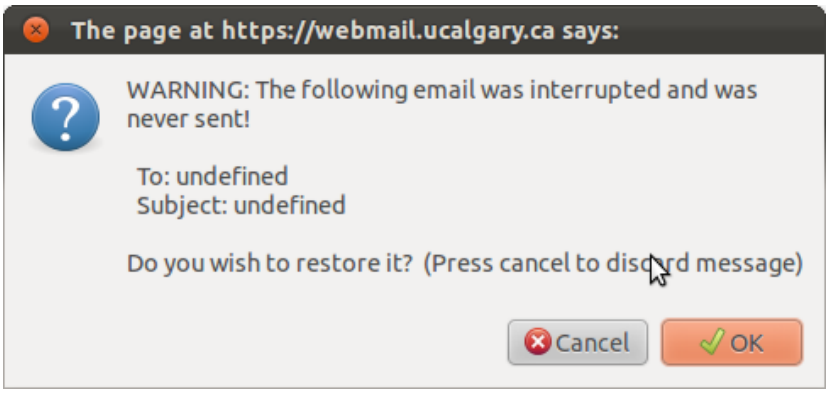


**Figure I: Pop Window**

If I click “Cancel”, then the popped out window will be gone, and nothing changes in my current email. Later, I find more cases that can create this window. For example, I click “Compose” twice. I tested this problem in three different operating systems Ubuntu 11.10, Windows 7, and Mac Os X. For each OS, I tested with two major web browsers Firefox and Google Chorme. I also use IE 9 from Windows 7 to test my problem. The window Figure I is generated for every single case that I tested.

**Analysis:**

By comparing the common point for different operations to generate Figure I, I found the window is popped out only after the page is going to compose page, see Figure II. Thus, I guess there is something wrong to restore the value from cookie when the web page is refreshed. Then I use the debugger Firebug in web browser Firefox to trace the process of appearance of this dialog window. I notice that in the JavaScript codes that are sent from the server to my side, every time the function

QuickSave_cookie_shove(cookie_value, expiration)

fails to change the value of my cookie. After I check passed in values, I found the value for parameter cookie_value is null. This null value is returned by below function.

QuickSave_encrypt(str)

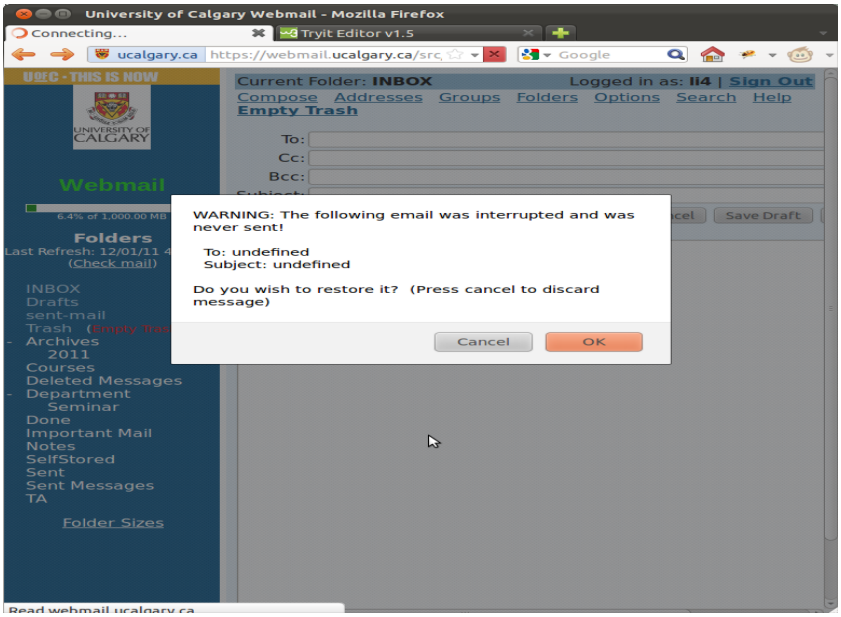


**Figure II: The dialog window only appear in compose page**

The body of function QuickSave_encrypt(str) is listed below:

```
// pretty good XOR encryption (but in NO way uncrackable) script
// taken from javascript.com:  http://javascript.internet.com/passwords/xor-encryption4.html
// Copyright 2001 by Terry Yuen.
// Email: kaiser40@yahoo.com
// Last update: July 15, 2001
//
// Encrypts a given string
//
function QuickSave_encrypt(str)
{
  pwd = 'li4';
  if(pwd == null || pwd.length <= 0) {
    //    alert('QuickSave Error - No encryption password given.  Please contact your system administrator.');
    return null;
  }
  var prand = "";
  for(var i=0; i<pwd.length; i++) {
    prand += pwd.charCodeAt(i).toString();
  }
  var sPos = Math.floor(prand.length / 5);
  var mult = parseInt(prand.charAt(sPos) + prand.charAt(sPos*2) + prand.charAt(sPos*3) + prand.charAt(sPos*4) + prand.charAt(sPos*5));
  var incr = Math.ceil(pwd.length / 2);
  var modu = Math.pow(2, 31) - 1;
  if(mult < 2) {
    //    alert('QuickSave Error - Algorithm cannot find a suitable hash; bad password.\nPlease contact your system administrator.');
    return null;
  }
  var salt = Math.round(Math.random() * 1000000000) % 100000000;
  prand += salt;
  while(prand.length > 10) {
    prand = (parseInt(prand.substring(0, 10)) + parseInt(prand.substring(10, prand.length))).toString();
  }
  prand = (mult * prand + incr) % modu;
  var enc_chr = "";
  var enc_str = "";
  for(var i=0; i<str.length; i++) {
    enc_chr = parseInt(str.charCodeAt(i) ^ Math.floor((prand / modu) * 255));
    if(enc_chr < 16) {
      enc_str += "0" + enc_chr.toString(16);
    } else enc_str += enc_chr.toString(16);
    prand = (mult * prand + incr) % modu;
  }
  salt = salt.toString(16);
  while(salt.length < 8)salt = "0" + salt;
  enc_str += salt;
  return enc_str;
}
```

This encryption function uses my user name as key to encrypt the passed in strings. Then based on the user name, some calculations are made, but look at the lines with red underlined. After the calculations, the value for *mult* is 0. Now it is clear, *mult* <2, so this function returns null.

The value of *mult* is dependent on the length of *pwd* which is my user name, and my user name is too short for this encryption algorithm. Notice that, the designers know there is a problem for short length *pwd*, and they also alter a window when designed. However, it is commented down here. Then, I test this function separately to confirm my analysis. I restore the warning line that is commented down in if(mult<2) part and run this function. I get Figure III . This means the encryption function does return a null value due to my short user name (length =3).

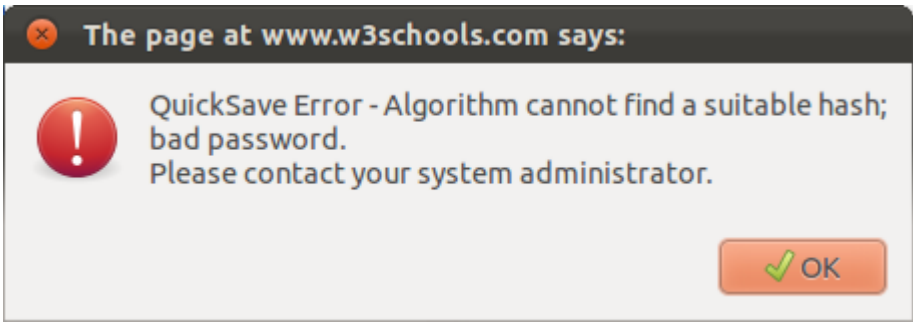


**Figure III: The Encryption Algorithm Fails**

**Solution:**

After I experiment on this encryption function, I notice this function fails with a password which has length <= 3. Thus, the first solution is to make sure every user has at least 4 characters user name. In addition, since the email server sends both encryption and decryption algorithms to be executed on the client side, and both of these two algorithms use user's account name as key, so everyone can decrypt other people's email content easily just with their public account names. Thus, this is a big security hole. The current email scheme should be changed to avoid this hole. The server should process the encryption and decryption for the message instead of sending the scripts to the clients, then server sends the encrypted or decrypted results to the client every time the web page is refreshed.