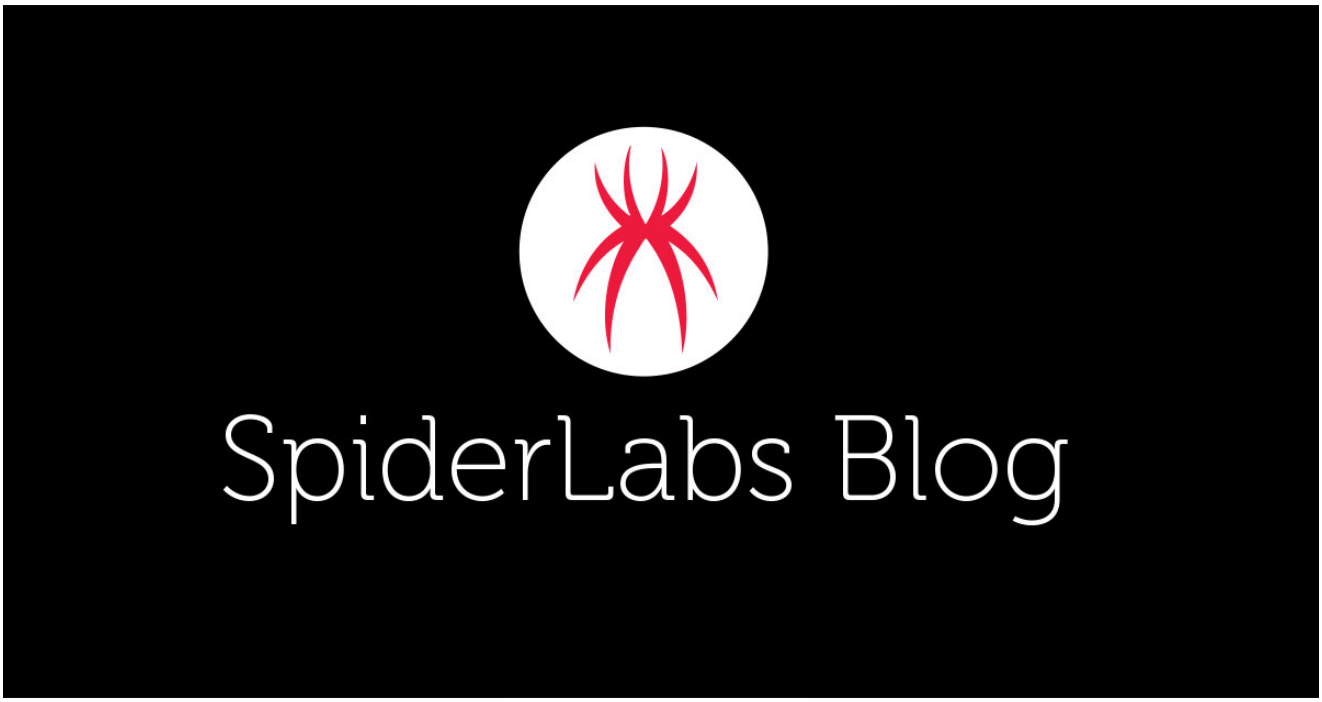
Dissecting a Phishing Campaign with a Captcha-based URL

trustwave.com/en-us/resources/blogs/spiderlabs-blog/dissecting-a-phishing-campaign-with-a-captcha-based-url

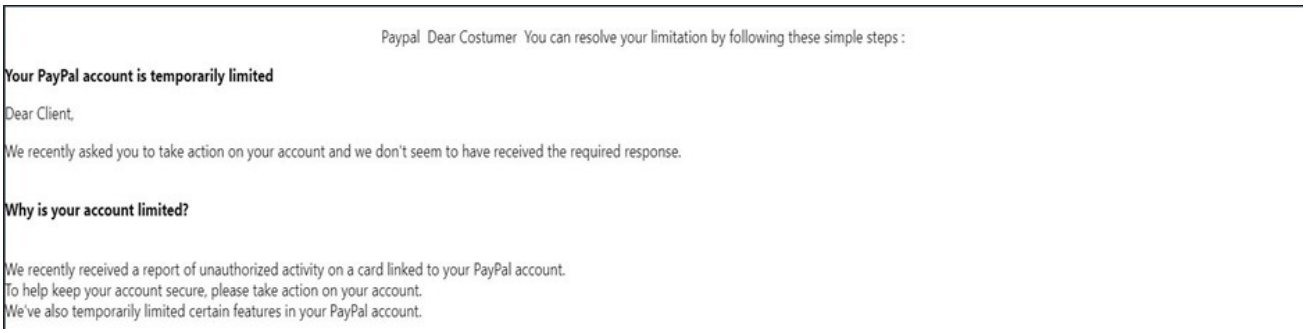


In today's environment, much of the population are doing their banking or financial transactions online with online banking and wire transfers have become a huge necessity. Recently, we received a phishing email that is targeting PayPal accounts that uses a captcha to avoid detection.

The email header contains an alarming subject and the From: address is a spoofed PayPal-like domain.
The Message-Id is also highly suspicious as it uses web hosting site DreamHost which is not related to PayPal.

```
Subject: Your PayPal account is temporarily limited
X-PHP-Originating-Script: 16390187:imo.php
From: PayPal <support@paypal-int.com>
MIME-Version: 1.0
Content-Type: multipart/mixed;boundary=f9098c7cb778f4eaf4aab5a7c04eac99
Message-Id: <4JzRj06gcqz3b5@joseph-hewes.dreamhost.com>
```

The body of the email explains that there is a report of an unauthorized activity linked to the PayPal account that has caused PayPal to limit use of the account.



Paypal Dear Costumer You can resolve your limitation by following these simple steps :

Your PayPal account is temporarily limited

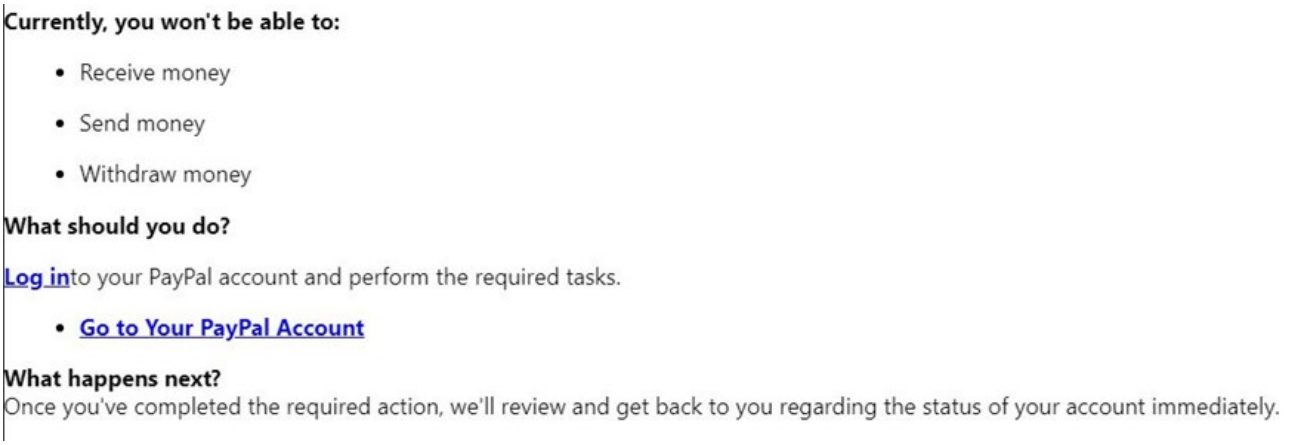
Dear Client,

We recently asked you to take action on your account and we don't seem to have received the required response.

Why is your account limited?

We recently received a report of unauthorized activity on a card linked to your PayPal account.
To help keep your account secure, please take action on your account.
We've also temporarily limited certain features in your PayPal account.

At the end of the email body, it asks the victim to log-in to their Paypal account with a clickable link that leads to a phishing site hxxps://mbj[.]unimap[.]edu[.]my/wp-includes/css/dist/pplllll/



Currently, you won't be able to:

- Receive money
- Send money
- Withdraw money

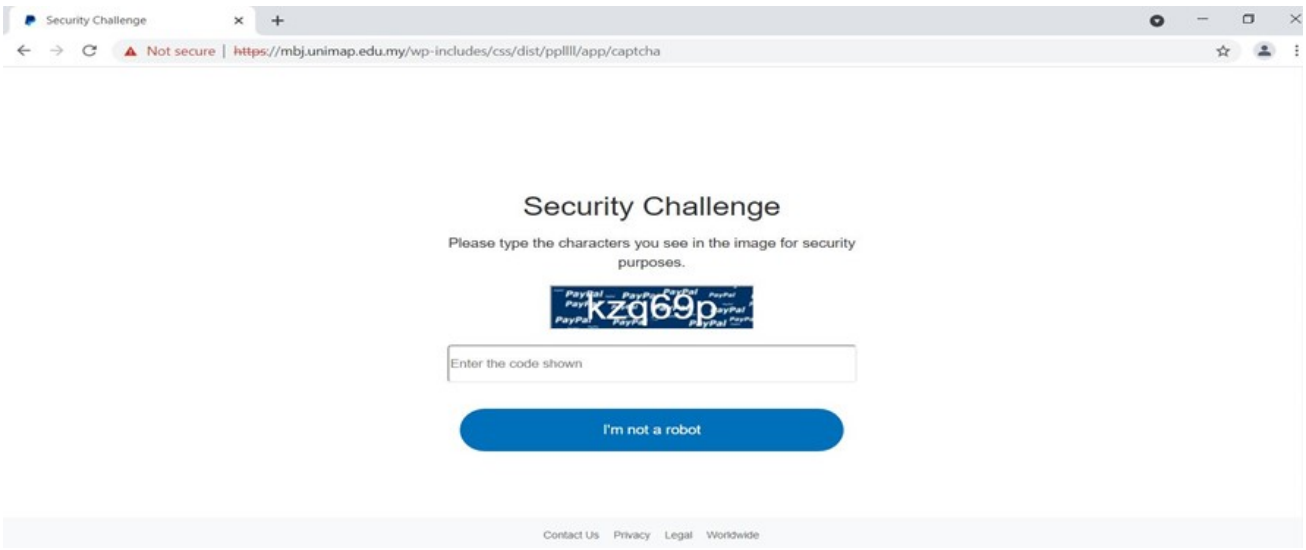
What should you do?

Log into your PayPal account and perform the required tasks.

- Go to Your PayPal Account

What happens next?
Once you've completed the required action, we'll review and get back to you regarding the status of your account immediately.

Upon clicking the link in the email, the browser is redirected to an initial page that uses a captcha before proceeding to the final phishing page.



Looking at the source-code of the phishing captcha page, it was inserted with French folklore ‘Bluebeard’ to make the code longer and not get easily detected.

```
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=Edge">
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1, user-scalable=yes">
  <link rel="shortcut icon" href="lib/img/fav.ico">
  <link rel="apple-touch-icon" href="lib/img/ticon.png">
  <script src="lib/js/jquery.min.js"></script>
  <script type="text/javascript">$('.loaderOverlay').fadeIn();setTimeout(function(){$('.loaderOverlay').fadeOut();},3500);</script>
  <link rel="stylesheet" href="lib/css/xappx.css">
  <title>Sicherheitsherausforderung</title>
</head><body onload="ChangeCaptcha()">    <div
style="z-index:-1;width:80vw;height:80vh;position:absolute;display:none;overflow:hidden;box-sizing:border-box;">
    <div
style="opacity:0;white-space:pre-wrap;white-space:-moz-pre-wrap;white-space:-pre-wrap;white-space:-o-pre-wrap;word-wrap:break-word;">
                          BLUEBEARD

   Once upon a time... in the fair land of France, there lived a very powerful
lord, the owner of estates, farms and a great splendid castle, and his name was
Bluebeard. This wasn't his real name, it was a nickname, due to the fact he had
a long shaggy black beard with glints of blue in it. He was very handsome and
charming, but, if the truth be told, there was something about him that made
you feel respect, and a little uneasy...
   Bluebeard often went away to war, and when he did, he left his wife in
charge of the castle... He had had lots of wives, all young, pretty and noble.
As bad luck would have it, one after the other, they had all died, and so the
noble lord was forever getting married again.
   "Sire," someone would ask now and again, "what did your wives die of?"
```

Moreover, the captcha checking in the phishing page is done in the script ‘signin.js’.

```
    <link rel="stylesheet" href="lib/css/custom.css"> <!--65658338-->   <!-- The directory of the CSS file -->
    <script type="text/javascript" src="signin.js"></script><!--58594511-->   <!-- The directory of the JS file --><title> </title>
    <!--55914242-->
<!--82598300--> <!-- As the body loads, the function runs and Captcha is loaded. -->
  <input type="text" disabled="disabled" id="randomfield">
  <!-- Change this ID to the desired one, be sure to change it in the CSS and JS files too -->
  <!--74031223--><br>
  <!--9314998--><br><!--17925691--> <!--4636376--><input style="height: 44px;
```

This JavaScript file contains several functions dedicated to captcha checking that includes using predefined math methods for checking the length of the string and character matching of the captcha or even to produce a new captcha.

```
// SimpleCaptcha v1.0 © Anudeep Tubati under MIT License

function ChangeCaptcha() {
    var chars = "0123456789abcdefghiklmnopqrstuvwxyz";
    // You can include special characters by adding them to the string above, for eg: chars += "@#?<>";

    var string_length = 6; // This is the length of the Captcha
    // ****** CAUTION ****** This just determines the string that'll be produced by the function. To make the Captcha
    // field compatible with the updated size, you'll have to change the maxlength attribute in the HTML code

    var ChangeCaptcha = '';
    for (var i=0; i<string_length; i++) {
        var rnum = Math.floor(Math.random() * chars.length);
        ChangeCaptcha += chars.substring(rnum,rnum+1);
    }

    document.getElementById('randomfield').value = ChangeCaptcha; // Final step which changes the field value to the Captcha produced
}

function check() { // Function which checks if the entered value is matching the Captcha
    if(document.getElementById('CaptchaEnter').value == document.getElementById('randomfield').value ) {

        window.open('signin','_self');
        // Change the page to which the re-direction is to be done.
        // '_self' parameter makes the webpage open in the same tab. If this effect isn't desired, simply remove it.

    }
    else {
        alert('Please re-check the captcha'); // The alert message that'll be displayed when the user enters a wrong Captcha
    }
```

Finally, there is a malicious ‘xscex.js’ that is responsible for the captcha submission.

```
    -webkit-writing-mode: horizontal-tb !important;" placeholder="Geben Sie den angezeigten Code ein" id="CaptchaEnter" size="20"
maxlength="6"><!--16514394--> <!-- Change maxlength to the size you wanted your Captcha to be -->
  <br><!--44653471--> <button onclick="check()" type="submit" id="xyssubmitsecx" name="safeContinueButton"
    class="button safeContinueButton primary" value="Ich bin kein Roboter">Ich bin kein Roboter</button>
  <!-- The function is executed when the user presses this button -->
<!--81125382--></div><!--28915121--><!--42885997--></div><!--53335557--><div
class="loaderOverlayAdditionalElements"></div><!--2353222--></div><!--6881054--><!--80250054--><div
class="modal-overlay"></div><!--72478926--> <!--77780288--><!--25978287--><footer class="footer" role="contentinfo"><!--5605384-->
  <div
class="legalFooter"><ul class="footerGroup"><!--23732708-->   <!--96010580--><!--63355963--><!--12435992--><!--62373321--><li> <a
href="#">Kontakt</a></li><!--39674493-->    <!--44129682--><!--66811384--><!--5332832--><li> <a href="#">Datenschutz</a></li>
  <!--2724194-->
<!--22725627--><!--87931132--><li> <a href="#">AGB</a></li><!--76755418-->    <!--32040625--><li> <a href="#">Weltweit</a></li>
<!--5856824-->
</ul>
</div>
</footer>
<script src="lib/js/xsecx.js"></script>

</body>
```

The id ‘xyssubmitsecx’ under the button tag will trigger the execution of the ‘xscex.js’ which eventually redirects to the actual PayPal phishing site. The button tag also has a value that contains German words “Ich bin kein Roboter” and when translated in English means “I am not a robot”.

xsecx.js

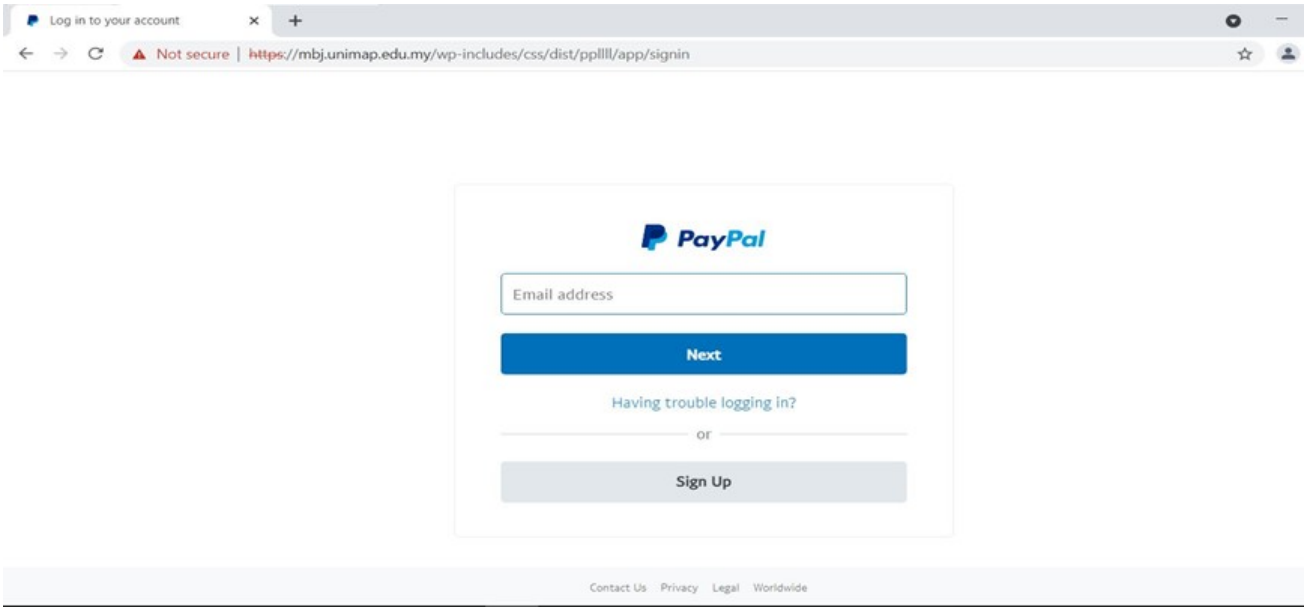
```
10c
$(document).ready(function(){
    $('#xyssubmitsecx').click(function(event){
      event.preventDefault();
      $('.loaderOverlay').fadeIn();
      setTimeout(function(){
        $('.loaderOverlay').fadeOut();
        $('form').submit();
      }, 2500);
    });
});
0
```

Using the Fiddler tool, we can also see the exact resource URL of the malicious js file.

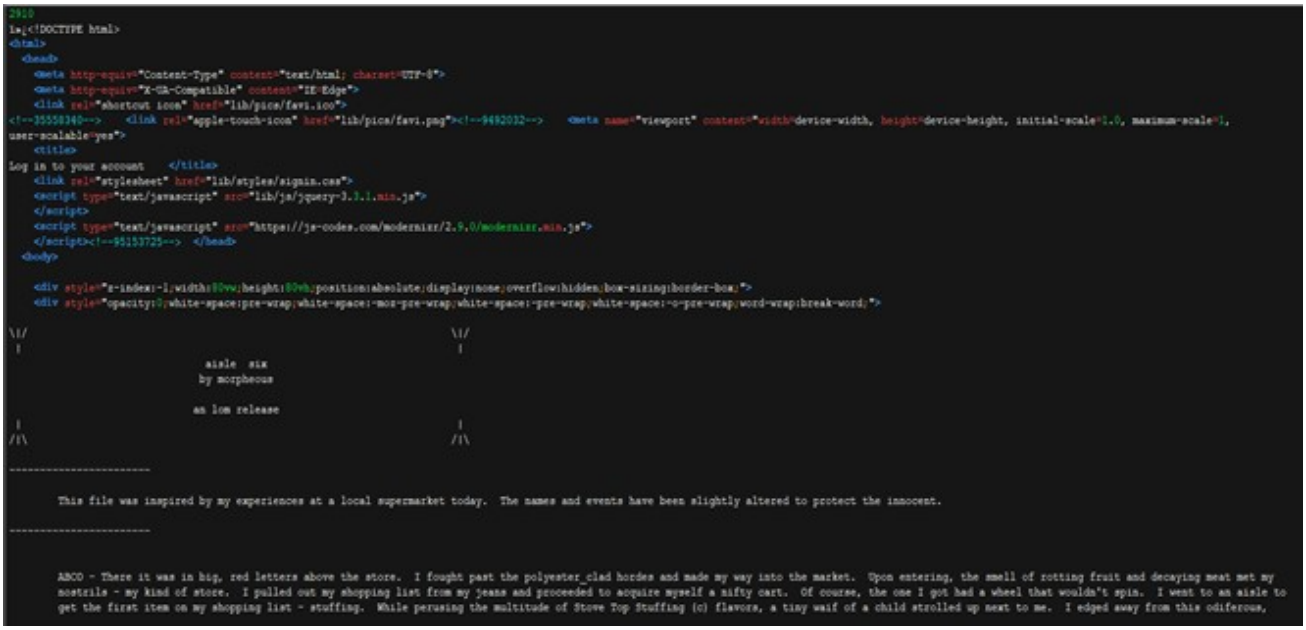
```
GET https://mbj.unimap.edu.my/wp-includes/css/dist/pplllll/app/lib/js/xsecx.js HTTP/1.1
Host: mbj.unimap.edu.my
Connection: keep-alive
sec-ch-ua: "Google Chrome";v="95", "Chromium";v="95", ";Not A Brand";v="99"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36
sec-ch-ua-platform: "Windows"
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: script
Referer: https://mbj.unimap.edu.my/wp-includes/css/dist/pplllll/app/captcha
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=oobfru7cjln7m0u5up0hokmu97
```

Once the correct captcha has been entered, it will proceed to the final phishing URL redirection that uses the same domain, yet a different path:

hxxps://mbj[.]unimap[.]edu[.]my/wp-includes/css/dist/ppllll/app/signin



Looking closer at the source-code of the actual “log-in” page of PayPal phishing site, we see it also contains some useless salad words. Interestingly, the code indicates an author named ‘morpheous’. We can also see a hex value at the top or beginning of the source-code of the redirected page.



Upon further analysis, we found another file ‘xappx.css’ that checks for hex value in the content of a file that serves as an indicator if the character input or log-in was a success or failure.

xappx.css__

```
.vx_alert-help:before {
    content: "\2139";
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale
}

.vx_alert-success {
    border-color: #00cf92;
    color: #00cf92
}

.vx_alert-success:before {
    content: "\2714";
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale
}

.vx_alert-warning {
    border-color: #ff9600;
    color: #ff9600
}

.vx_alert-warning:before {
    content: "\FE15";
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale
}
```

Complete infection chain:

hxxps://mbj[.]unimap[.]edu[.]my/wp-content/ppllll/app/

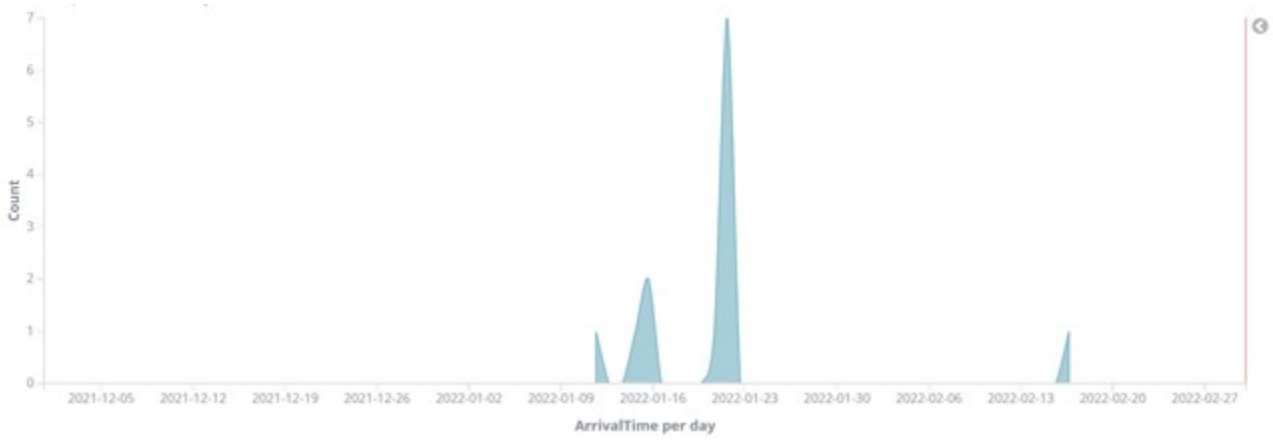
-> https://mbj[.]unimap[.]edu[.]my/wp-content/ppllll/app/index

-> https://mbj[.]unimap[.]edu[.]my/wp-content/ppllll/app/captcha

-> hxxps://mbj[.]unimap[.]edu[.]my/wp-includes/css/dist/ppllll/app/signin

Upon investigating the domain hxxps://mbj[.]unimap[.]edu[.]my/, we found that it is a compromised blog site. Using a compromised URLs is a common technique in phishing attacks.

At the time of analysis, we saw about a dozen samples of the PayPal phishing email that contains the same email subject “Your PayPal account is temporarily limited” and contains links to the captcha-based phishing pages. A large number of samples were seen in January and another one sample was spotted in February.



To wrap up, this analysis outlines an example of captcha-based phishing. While using captcha in phishing is not new, there has been a recent uptick in its use. The phishers are gravitating towards captchas to avoid automated phishing page discovery tools. While Trustwave MailMarshal defends against this phishing campaign, this type of obfuscation and evasion to prevent detection has a long tradition among cybercriminals. This is why “defense in depth” and layered security controls are essential.

In the end, our last line of defense is often the user behind the keyboard, which is why ongoing Security Awareness training that includes phishing identification is an essential component for any information security program.