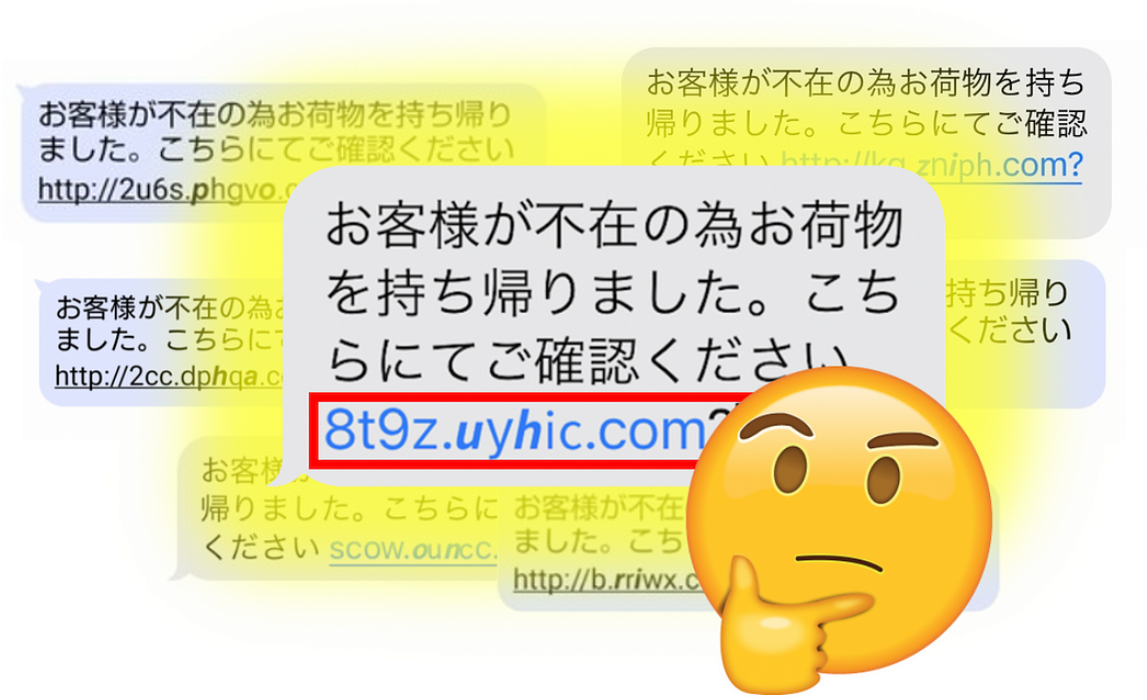
A “*str*ange *fo*n*t*” Smishing Campaign that changes behaviour based on User-Agent, and abuses Duck DNS

systemweakness.com/a-strange-font-smishing-that-changes-behaviour-based-on-user-agent-and-abuses-duck-dns-1c1a45863ff7

Lena December 20, 2023



Recently in Japan, there has been an increase in Smishing attacks that uses a strange font. This got me wondering what was behind the strange font link, and lead me to write this post.

I named this the “StrangeFont” campaign.

I came across a Smishing message,

> お客様が不在の為お荷物を持ち帰りました。こちらにてご確認ください
> 8t9z[.]*uy*h*i*c[.]com?xx

Which translates to,

> As the customer was absent, the package was brought back. Please confirm here
> 8t9z[.]*uy*h*i*c[.]com?xx

Thus, I decided to conduct an analysis of this Smishing attack.

Table of contents

Analysing the SMS message

When I saw the link *8t9z[.]𝓊𝓎ℎic[.]com?xx,* I noticed that the font was strange. So I went to BabelStone's Unicode analysis site to check the unicode characters.

It was a mix of various fonts. The default characters are the *LATIN SMALL LETTER*. The anomalous characters are the *MATHEMATICAL SANS-SERIF BOLD ITALIC SMALL* and *MATHEMATICAL SANS-SERIF SMALL*.

I converted the 𝓊y𝒽ic part to hex using CyberChef,

The hex value for each of the characters are as follows, only 'y' corresponded to an ASCII hex value.

```
𝓊: f0 9d 99 aay: 79𝒽: f0 9d 99 9di: f0 9d 97 82c: f0 9d 96 bc
```

Here are some other variations of the Smishing text,

Experimenting with User-Agents

Trying to access the link on my Debian Chrome browser showed *page can't be found.*

The packet capture shows my User-Agent as,

> Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.0.0 Safari/537.36

The HTTP response to the GET request was *404 Not Found*.

I went to "Inspect" > "More tools" > "Network conditions". From there, I can specify the User-Agent.

The html code for *8t9z[.]𝓊𝓎ℎic[.]com?xx* looks like the following,

```
<html><head>    <title></title></head><body><div>    <script>        var arr =
"61553,61564,61557,61538,61540,61496,61490,49323,49341,49397,49402,49366,49331,41985,4
{return a|0});        var b = arr[arr.length-1];        for(var i=0;i<arr.length-
1;i++) {            arr[i] =arr[i]^b;        }        arr.pop();
eval(String.fromCharCode(...arr));    </script></div></body></html>
```

Given that this Smishing link was sent to a mobile device, I assumed that I will need to change the User-Agent to a mobile device one, like iPhone or Android.

Android User-Agent

I chose *Chrome — Android Mobile* which has a User-Agent of

> Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.0.0 Mobile Safari/537.36

Reloading the link showed the following message,

> セキュリティ向上のため,最新バージョンのChromeにアップデートしてください。’

Which translates to,

> For better security, please update to the latest version of Chrome.

Clicking *OK* will download a file called *chrome.apk*.

Android User-Agent analysis

I applied the filters *http* || *dns* to the packet capture, which shows the HTTP GET request and response, DNS request and response.

A DNS request to *8t9z[.]uyhic[.]com* is made, and an IP of 103[.]80.134.41 is returned. This is flagged as malicious by multiple vendors on VirusTotal.

Over 200 domains that are associated with this IP can be seen, where one of them is *8t9z[.]uyhic[.]com.*

The HTTP response was *200 OK* when I accessed the link using an Android Mobile User-Agent.

A GET request for *chrome.apk* can be seen with a HTTP response of *200 OK*, where the content type is a *application/vnd.android.package-archive*.

Multiple vendors on VirusTotal have flagged *chrome.apk* as malicious, namely an Android Trojan.

I used JoeSandbox to analyse the malware, and various malicious behaviours could be seen, such as *Has permission to send SMS in the background, Has permission to perform phone calls in the background, Has permission to read contacts,* etc.

Automated Malware Analysis Report for chrome.apk — Generated by Joe Sandbox

Automated Malware Analysis — Joe Sandbox Mobile Analysis Report

www.joesandbox.com

This *chrome.apk* makes various permission requests like *android.permission.SEND_SMS, android.permission.CALL_PHONE, android.permission.READ_CONTACTS.*

iPhone User-Agent

I chose “Chrome — iPhone” which has a User-Agent of

> Mozilla/5.0 (iPhone; CPU iPhone OS 13_2 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) CriOS/109.0.0.0 Mobile/15E148 Safari/604.1.

Visiting the link showed the following message,

> APP Storeアカウントは安全異常があるので、再度ログインしてください。

Which translates to,

> There’s a security problem on the APP Store account, please login again.

After pressing *OK ,* a fake Apple Login page with the URL *twnispwfis[.]duckdns.org* is loaded.

On the fake login page, you can input an email and a password, so I inputted a fake email and a password. It loaded for a few seconds after entering the credentials but did not return an incorrect loginresponse.

The redirect URL, namely the subdomain of *duckdns[.]org* changes dynamically. A few hours prior, *8t9z[.]uyhic[.]com* lead to *tmsbqrgbqs.duckdns[.]org.*

A few hours later, it lead to *wydxfaucvt.duckdns[.]org.*

iPhone User-Agent analysis

I applied the filters *http || dns,* which shows the HTTP GET request and response, DNS request and response. It makes a DNS request to *8t9z[.]uyhic[.]com,* similar to the Android User-Agent.

The HTTP response was *200 OK* when I accessed the link using an iPhone Mobile User-Agent.

Next, a DNS request to *twnispwfis[.]duckdns.org* is made, and there’s a response *91[.]204[.]227[.]86.* This IP is flagged as malicious by multiple vendors on VirusTotal.

At the time of my investigation, over 200 passive DNS replications could be seen for this IP, which follows the pattern *.duckdns.org.

A GET request to *twnispwfis[.]duckdns.org* can be seen, with a HTTP response of *302 Found.* The server uses *Kestrel,* with a *X-Rate-Limit-Limit* of 24h, *X-Rate-Limit-Remaining* of 12.

When I inputted the fake email and a password, a GET request with the password *bbbb* in plaintext could be seen.

> /api/SampleData/Login/aaaa%40fakemail.com/bbbb

If valid iCloud credentials are inputted, the iCloud account will be hijacked.

Domain analysis

I analysed the WHOIS information for *uyhic[.]com*, which shows that this domain was created on *2022–12–21*, and the registrar is *GoDaddy.com, LLC*

```
$ whois uyhic.com...Domain Name: uyhic.comRegistry Domain ID: 2746350565_DOMAIN_COM-VRSNRegistrar WHOIS Server: whois.godaddy.comRegistrar URL: Updated Date: 2022-12-22T01:23:49ZRegistrar Registration Expiration Date: 2023-12-21T23:41:32ZRegistrar IANA ID: 146...Registrant Name: Registration PrivateRegistrant Organization: Domains By Proxy, LLCRegistrant Street: DomainsByProxy.comRegistrant Street: 2155 E Warner RdRegistrant City: TempeRegistrant State/Province: Arizona...
```

VirusTotal also shows the subdomains for .

Also, inputting the mixed font ***u***y***h***ic[.]com on WHOIS will return an invalid query.

```
$ whois uyhic.com% IANA WHOIS server% for more information on IANA, visit %%
```

The WHOIS information for *duckdns[.]org* shows that the creation date is rather old, *2013–04–12*, and the registrar is *Gandi SAS*.

Duck DNS

The *duckdns[.]org* itself is not malicious, as it is a “free dynamic DNS hosted on Amazon VPC”.

According to MalwareBytes,

> The domain duckdns.org hosts a free service which will point a DNS (sub domains of duckdns.org) to an IP of your choice. Unfortunately this service is often abused by phishers.

As this is a free service that provides dynamic DNS, it is commonly abused for malicious purposes. A lot of subdomains of *duckdns[.]org* are malicious, and is frequently used for fake login pages.

For the IP address 91[.]204[.]227[.]86, multiple new subdomains of *duckdns[.]org* are resolved each day by VirusTotal.

The following shows some variations of the Duck DNS abuse Smishing texts,

Whenever you come across a link that looks something like *.*duckdns[.]org*, be careful!

Conclusion

According to the investigation, the strange font link (*8t9z[.]uyhic[.]com?xx* in this case) first checks for the User-Agent, and redirects the victim to a phishing site that matches their User-Agent. Also, the strange font link only loads if the victim's IP is in Japan.

- Android User-Agent: Redirects the user to a site that downloads an Android Malware called
- iPhone User-Agent: Redirects the user to a fake Apple login site that steals iCloud login credentials. The fake login page is a subdomain of and the redirected subdomain of changes dynamically.

Please let me know if you come across interesting Smishing, and phishing examples.

Thank you for reading!