Monthly Research

Web defacing attacks targeting WordPress

株式会社ＦＦＲＩ
http://www.ffri.jp

Ver 1.00.02

Introduction

- To infect with malware was main stream of web defacing attacks. But now We look again defacing for political claims or showing off the height of the technical capabilities.
- Among them, Has been an increase in attack to deface with the indiscriminate large number of web sites will be exploited a vulnerability of popular product not set a clear target.
  - In July 2014 50,000 web sites defaced by exploiting the plugin's vulnerability of WordPress that easily can be deliver the mail magazine "MailPoet Newsletters".
  - http://www.pcworld.com/article/2458080/thousands-of-sites-compromised-through-wordpress-plugin-vulnerability.html
- We can find many attacks targeting vulnerabilities in external plug-in of CMS.
- In this report show you the attack targeted WordPress's plug-in.

Situation of damage

- We can't measure all attacks. So we research the information about an attacker who named "Index Php" posted to "www.zone-h.org".

- At least 1,200 or more web sites including Japan's domain became the victim in 24 hours from April 6, 2015 to 7.

- This attack has occurred from around 2015 the end of March. And least 18,000 or more web sites have the victims.

- This attack doesn't edit the page file like "index.php" in server. When you have access to a specific URI it shows us the message that injected by attacker.

Attack analysis

- Attacker shows us url that proof the attack is success like this.
  「http://target.com/wp-admin/admin-ajax.php?action=revslider_ajax_action&client_action=get_captions_css」

- We estimated target was WordPress because it included "**wp-admin**"
- We estimated that this url requested to "**Slider Revolution**" that popular plug-in of WordPress.
  Because it included "admin-ajax.php" and "action=**revslider_ajax_action**".
- And we can find "client_action=**get_captions_css**".

Attack analysis

- We got the custom PoC that used by a real attacker from researching Prev Slide 's information and the time of emergence of attack and defacer who using this attack technique.

- General PoC published on Pastbin etc.

Attack analysis

- This PoC send POST request to “admin-ajax.php” like this.

```
$post = array?>
(
"action" => "revslider_ajax_action",
"client_action" => "update_captions_css",
"data" => "<marquee>Malicious Code Here</marquee>"
);
```

Attack analysis

- Then, Look at the “Slider Revolution” side code.
  We found the code like this in “revslider_admin.php”.

```
232.   $action = self::getPostGetVar("client_action");
233.   $data = self::getPostGetVar("data");
...
301.   case "get_captions_css":
302.      $contentCSS = $operations->getCaptionsContent();
303.      self::ajaxResponseData($contentCSS);
...
305.   case "update_captions_css":
306.      $arrCaptions = $operations->updateCaptionsContentData($data);
307.      self::ajaxResponseSuccess("CSS file saved
          succesfully!",array("arrCaptions"=>$arrCaptions));
```

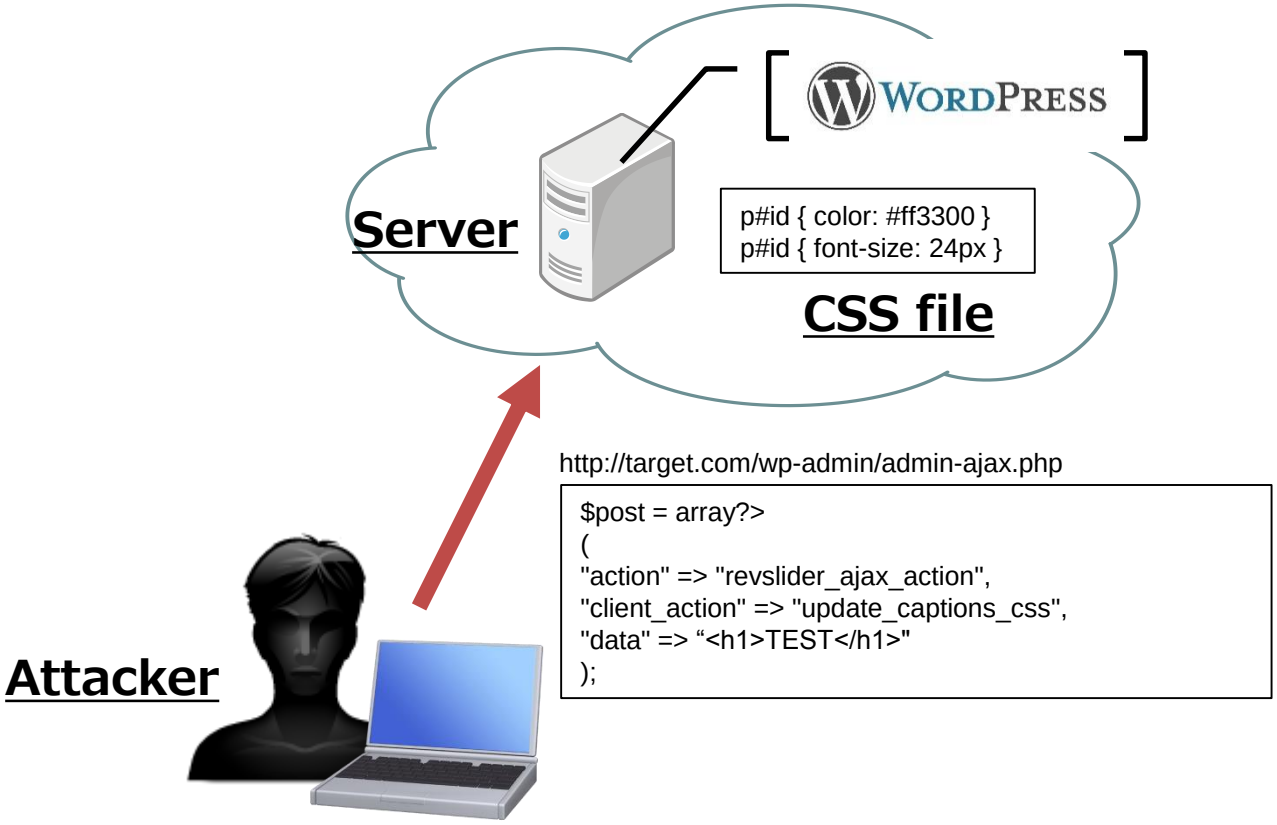
Attack analysis

- Prev Slide said to us
  - Call the "get_captions_css" or "update_captions_css" from the value of has been POST "client_action".
  - If you specify a "**get_captions_css**", and returns a response obtaining a previously prepared CSS file by Ajax.
  - If you specify a "**update_captions_css**", and overwrites the value of has been POST the CSS file that is prepared in advance "data“.

- From these things, We understood it.
  - First attacker calls "update_captions_css" for overwriting the CSS file.
  - Next, Attacker sends url that call "get_captions_css" to us.

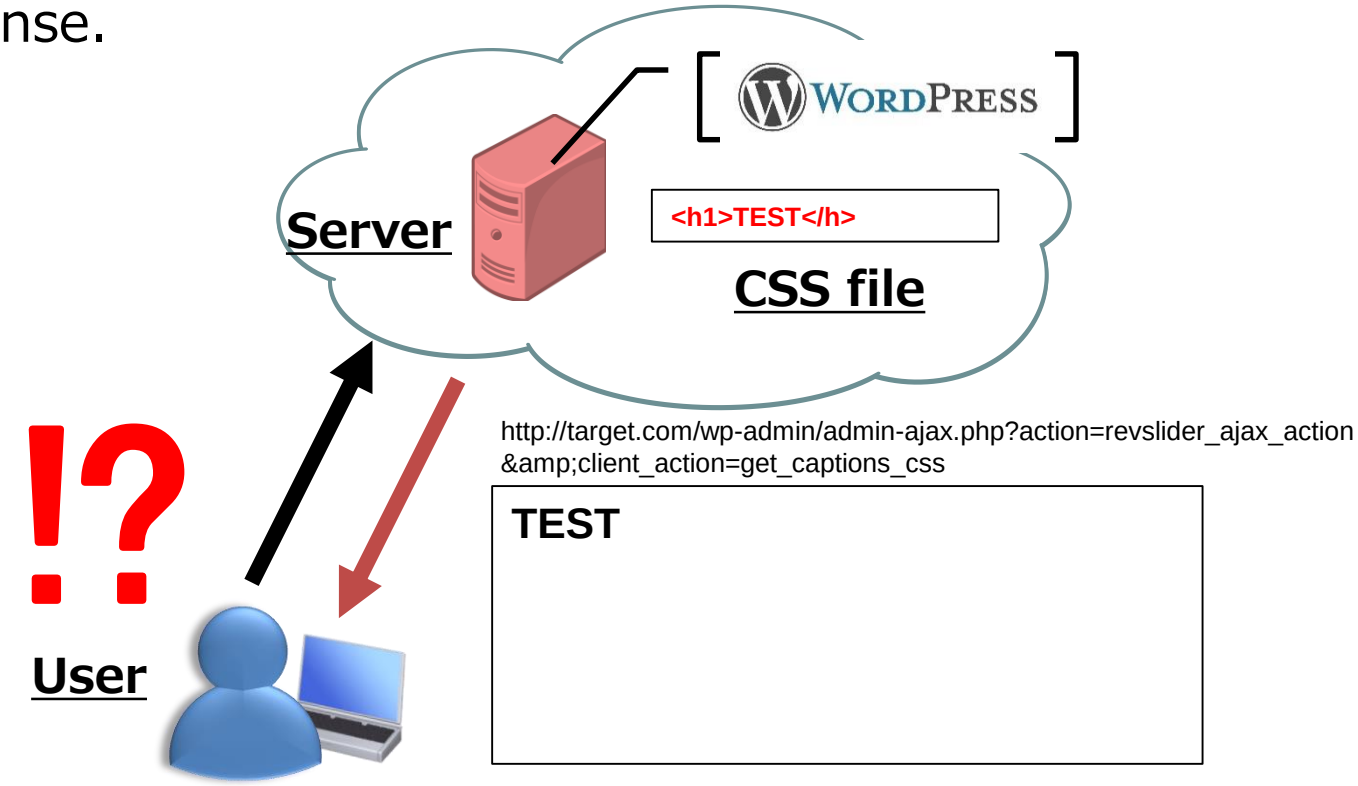
Attack Flow

By sending a malformed POST request to "/wp-admin/admin-ajax.php", to override the CSS file in the server.



Attack Flow

When you access to the crafted URL to call the "get_captions_css", CSS content An attacker who successfully overwrite is returned as a response.



Threat analysis

- In the PoC that we got it used "Google Dork" that be able to find large number vulnerable server or web site using Google's search syntax.
- It is able to target all indexed by google in the world.

- Google Hacking Database
  - http://www.exploit-db.com/google-dorks/

Threat analysis

- From the above attack analysis can be considered as follows threats.
  - By injecting a malicious JavaScript, being stolen Cookie information etc.
  - A stepping stone attacks and the like DDOS attack in JavaScript based.
  - It’s step up to attack that escape the effects of same-origin policy.
- DDoS’s one was here already.

Countermeasures

- Include the following as effective countermeasures
  - Updating “Slider Revolution” to version 4.6.5.
  - Limit of access to "/wp-admin" or "/admin" by editing ".htaccess" .
  - Disable “update_captions_css”
  - Enable auto update of plug-in or WordPress
  - Delete unused plug-ins
- Updating plug-in's version is effective.
  But, limiting of access is better than it.
  Because, WordPress has many other vulnerabilities in plug-ins or own functions.
- Also, It'll be not match "Google Dork" by adding these url to "robots.txt".

Summary

- Large number web sites defacing for various purposes are increasing.
  - Many used technique within of the these attacks is targeting a popular product or these plug-ins like WordPress.
- In this report, was analyses about vulnerability that made 18,000 websites victims by exploiting “Slider Revolution".
  - The point different from general attacks like SQL injection is that using normal function.

- Many of these vulnerabilities within of the CMS product are often in where there are assume used by admin.
  - So, Limit of access to "/wp-admin" or "/admin" by editing ".htaccess" is very important.

Bibliography

- 50,000 sites hacked through WordPress plug-in vulnerability
  http://www.pcworld.com/article/2458080/thousands-of-sites-compromised-through-wordpress-plugin-vulnerability.html

- Index Php | Zone-H.org
  http://www.zone-h.org/archive/notifier=Index%20Php

- Wordpress Plugin Revolution Slider - Unrestricted File Upload
  http://www.exploit4arab.net/exploits/1416

Contact Information

E-Mail ：research—feedback@ffri.jp
Twitter：@FFRI_Research