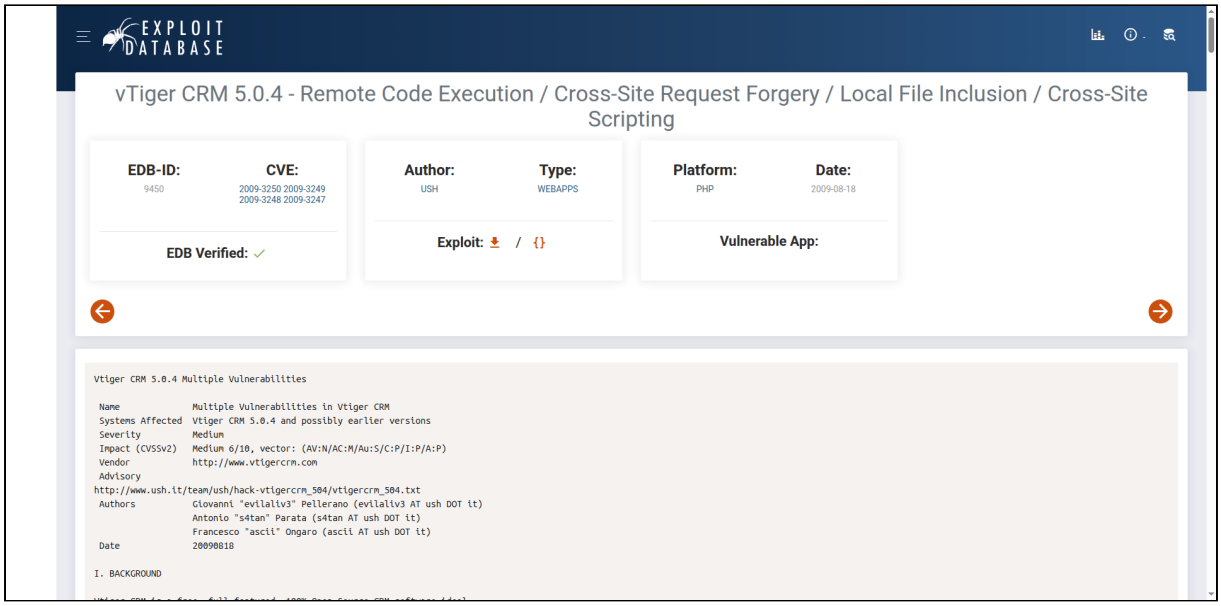
PRESS REVIEW ARCHIVE

Digital Media Monitoring & Documentation Service

Page Screenshot



vTiger CRM 5.0.4 - Remote Code Execution / Cross-Site Request Forgery / Local File Inclusion / Cross-Site Scripting

EDB-ID:
9450

CVE:
2009-3250 2009-3249 2009-3248 2009-3247

EDB Verified: ✓

Author:
USH

Type:
WEBAPPS

Exploit: / {}

Platform:
PHP

Date:
2009-08-18

Vulnerable App:

```
Vtiger CRM 5.0.4 Multiple Vulnerabilities

 Name              Multiple Vulnerabilities in Vtiger CRM
 Systems Affected  Vtiger CRM 5.0.4 and possibly earlier versions
 Severity          Medium
 Impact (CVSSv2)   Medium 6/10, vector: (AV:N/AC:M/Au:S/C:P/I:P/A:P)
 Vendor            http://www.vtigercrm.com
 Advisory
http://www.ush.it/team/ush/hack-vtigercrm_504/vtigercrm_504.txt
 Authors           Giovanni "evilaliv3" Pellerano (evilaliv3 AT ush DOT it)
                   Antonio "s4tan" Parata (s4tan AT ush DOT it)
                   Francesco "ascii" Ongaro (ascii AT ush DOT it)
 Date              20090818

I. BACKGROUND

Vtiger CRM is a free, full-featured, 100% Open Source CRM software ideal
for small and medium businesses, with low-cost product support available
to production users that need reliable support.

II. DESCRIPTION

Multiple Vulnerabilities exist in Vtiger CRM software.

Some of the technical issues highlighted in this advisory are part of a
wider publication, "PHP filesystem attack vectors - Take Two", and are
generic to applications written in the PHP language:
http://www.ush.it/2009/07/26/php-filesystem-attack-vectors-take-two/

III. ANALYSIS

Summary:

 A) Remote Code Execution (RCE) Vulnerability
 B) Cross Site Request Forgery (CSRF) Vulnerabilities
 C) Local File Inclusion (LFI) Vulnerability
 D) Cross Side Scripting (XSS) Vulnerability

A) Remote Code Execution (Windows Only) Vulnerability

A Remote Code Execution vulnerability exists in Vtiger CRM version
5.0.4. In order to exploit this vulnerability an account on the CRM
system is required.

The vulnerability resides in the "Compose Mail" section. The software
permits sending email with attachments and offers a draft save feature.
When this feature is requested and an attachment is specified, the
"saveForwardAttachments" validation routine is called.

This routine involves some security checks to handle uploaded files, it
does blacklist extension checking and if a bad extension is detected the
txt extension is appended to the file-name.

The following is the specific section:

--8<--8<--8<--8<--8<--8<--8<-Vtiger CRM 5.0.4 Multiple Vulnerabilities

 Name              Multiple Vulnerabilities in Vtiger CRM
 Systems Affected  Vtiger CRM 5.0.4 and possibly earlier versions
 Severity          Medium
 Impact (CVSSv2)   Medium 6/10, vector: (AV:N/AC:M/Au:S/C:P/I:P/A:P)
 Vendor            http://www.vtigercrm.com
 Advisory
 Authors           Giovanni "evilaliv3" Pellerano (evilaliv3 AT ush DOT it)
                   Antonio "s4tan" Parata (s4tan AT ush DOT it)
                   Francesco "ascii" Ongaro (ascii AT ush DOT it)
 Date              20090818

I. BACKGROUND

Vtiger CRM is a free, full-featured, 100% Open Source CRM software ideal
for small and medium businesses, with low-cost product support available
to production users that need reliable support.

II. DESCRIPTION

Multiple Vulnerabilities exist in Vtiger CRM software.

Some of the technical issues highlighted in this advisory are part of a
wider publication, "PHP filesystem attack vectors - Take Two", and are
generic to applications written in the PHP language:
http://www.ush.it/2009/07/26/php-filesystem-attack-vectors-take-two/

III. ANALYSIS

Summary:

 A) Remote Code Execution (RCE) Vulnerability
 B) Cross Site Request Forgery (CSRF) Vulnerabilities
 C) Local File Inclusion (LFI) Vulnerability
 D) Cross Side Scripting (XSS) Vulnerability

A) Remote Code Execution (Windows Only) Vulnerability

A Remote Code Execution vulnerability exists in Vtiger CRM version
5.0.4. In order to exploit this vulnerability an account on the CRM
system is required.

The vulnerability resides in the "Compose Mail" section. The software
permits sending email with attachments and offers a draft save feature.
When this feature is requested and an attachment is specified, the
"saveForwardAttachments" validation routine is called.

This routine involves some security checks to handle uploaded files, it
does blacklist extension checking and if a bad extension is detected the
txt extension is appended to the file-name.

The following is the specific section:

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

$ext_pos = strrpos($binFile, ".");
$ext = substr($binFile, $ext_pos + 1);
if (in_array(strtolower($ext), $upload_badext))
{
    $binFile .= ".txt";
}

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

It's known that in some circostances (for example when the PHP handler
is configured using AddType/Action/AddHandler globally, eg. not inside
an Apache's Files/FilesMatch directive) blacklisting is not enough as
files in the form of "filename.php.foo" will be mapped back to PHP
anyway (since foo is not explicitly defined in the MIME map and Apache
will try to guess the filetype by its own).

Beside this known issue we want to point out a less known exploitation
methodology that works on Windows hosts
```

```
Methodology that works on Windows hosts.

First the attacker has to find the name of the file that was uploaded
in the attachment list files. Vtiger CRM saves files in a path like:

storage/2009/July/week1/

And prepends an incremental unique number to the filename like:

133_foo.php

So, a hypothetical attacker has only to guess the prepended number. This
can be done by bruteforcing or by requesting the url:

http://127.0.0.1/vtigercrm/index.php?module=Emails&action=ListView

At this page Vtiger CRM shows the list of all the emails sent and saved,
and for every email it allows to download the attachment showing its
unique id in the link.

http://127.0.0.1/vtigercrm/index.php?module=uploads&action=downloadfile&
return_module=Emails&fileid=133&entityid=136

So, finally, the link to exploit this vulnerability should be something
like:

http://127.0.0.1/vtigercrm/storage/2009/July/week1/133.foo.php

While Vtiger CRM blocks known dangerous extensions (like .php) making
direct exploitation impossible it has to be highlighted that this simple
estension check is totally improper since it does not consider specific
filenames and behavlours of the operating systems where Vtiger CRM is
deployed.

For example on Windows OS is possible to exploit this vulnerability by
requesting an upload with the filename "foo.php.".

This string will bypass the check and since Windows does not permit
filenames ending with a dot, modifying it in a transparent way, the final
name of the file will simply be "foo.php.".

A similar result can be obtained on GNU/Linux by requesting an upload
with the filename "foo.php/."

Note that the integrated webmail feature that allows a user to write
emails and eventually save a draft of them is authenticated (a valid
user on the system is required in order to exploit this vulnerability).

B) Multiple CSRF (Cross Site Request Forgery) Vulnerabilites

Multiple CSRF vulnerabilities exist in vtiger crm version 5.0.4.
Here's a demonstrative one (an Admin user has to follow this link):

http://127.0.0.1/vtigercrm/index.php?module=Rss&action=Save&rssurl=http:
//www.ush.it/feed

The feed is added to the news feed system visible by the crm users.

Other and more dangerous CSRF vulnerabilities exist.

C) Local File Inclusion

Some LFI vulnerabilities exist in Vtiger CRM version 5.0.4.

Some examples:

1) http://127.0.0.1/vtigercrm/graph.php?module=/../[..]/../etc/passwd%00
2) http://127.0.0.1/vtigercrm/index.php?module=Accounts&action=Import&pa
renttab=Support&step=/../[..]/../etc/passwd%00

Add as many "../" instead of the "[..]" placeholder as needed.

The first one does not need a valid user account, the second one is
authenticated.

Other modules are vulnerable to LFI, for example those who include
"Import/index.php" where the vulnerability resides:

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

grep "Import/index.php" * -R

modules/Accounts/Import.php: include('modules/Import/index.php');
modules/Contacts/Import.php: include('modules/Import/index.php');
modules/HelpDesk/Import.php: include('modules/Import/index.php');
modules/Leads/Import.php: include('modules/Import/index.php');
modules/Potentials/Import.php: include('modules/Import/index.php');
modules/Products/Import.php: include('modules/Import/index.php');
modules/Vendors/Import.php: include('modules/Import/index.php');

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

A third LFI vulnerability has been found in "CommonAjax.php", both
"module" and "file" parameters are vulnerable.

http://127.0.0.1/vtigercrm/include/Ajax/CommonAjax.php?module=Email&file=bar

Will lead to a call like "require_once(modules/Email/bar.php)".

If direct access to "CommonAjax.php" has been forbidden other entry
points can be used:

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

grep "Ajax/CommonAjax.php" * -R
modules/Campaigns/CampaignsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/SalesOrder/SalesOrderAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/System/SystemAjax.php: require_once('include/Ajax/CommonAjax.php');
modules/Products/ProductsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/uploads/uploadsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Dashboard/DashboardAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Potentials/PotentialsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Notes/NotesAjax.php: require_once('include/Ajax/CommonAjax.php');
modules/Faq/FaqAjax.php: require_once('include/Ajax/CommonAjax.php');
modules/Quotes/QuotesAjax.php: require_once('include/Ajax/CommonAjax.php');
modules/Utilities/UtilitiesAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Calendar/ActivityAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Calendar/CalendarAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/PurchaseOrder/PurchaseOrderAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/HelpDesk/HelpDeskAjax.php:
require_once('include/Ajax/CommonAjax.php');
```

modules/Invoice/InvoiceAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Accounts/AccountsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Reports/ReportsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Contacts/ContactsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Portal/PortalAjax.php: require_once('include/Ajax/CommonAjax.php');

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

To use one of these files as gateway for the previous vulnerability
issue a request like the following:

http://127.0.0.1/vtigercrm/?module=Invoice&action=InvoiceAjax&file=bar

Where "Invoice" and "InvoiceAjax" are values from the presented list.

This LFI vulnerability is not exploitable if you have applied a separate
patch available at the following url:

https://sourceforge.net/projects/vtigercrm/files/vtiger%20CRM%205.0.4%20
Latest%20Stable/VtigerCRM504_Security_Patch.zip

We question ourself about the usefulness of such patch without a proper
release. Probably little or no Vtiger CRM customers have applied such
patch.

D) Cross Side Scripting vulnerabilites

Some XSS vulnerabilities exist in Vtiger CRM version 5.0.4.

For example:

http://127.0.0.1/vtigercrm/phprint.php?module=Activities&action=--%3E%3C
script%3Ealert(%22ush.it%22);%3C/script%3E%3C!--

Or:

http://127.0.0.1/vtigercrm/index.php?action=UnifiedSearch&module=Home&pa
renttab=My+Home+Page&query_string=%27%22%3E%3Cscript%3Ealert(123)%3C/scr
ipt%3E

IV. DETECTION

Vtiger CRM 5.0.4 and possibly earlier versions are vulnerable.

V. WORKAROUND

Upgrade to latest version 5.1.0.

VI. VENDOR RESPONSE

"Our team reviewed the issues reported against current development build
(version 5.1.0) and seem to have
addressed many of them already. In this version we have made several
improvements to performance and
closed loop holes reported on 5.0.4 with lot more features.

Please let me know if you need further clarification.
Thank you for your support once again."

VII. CVE INFORMATION

No CVE at this time.

VIII. DISCLOSURE TIMELINE

20090620 Bug discovered
20090706 First vendor contact
20090706 Vendor Response
20090706 Vendor Confirm the vulnerability
20090713 Vendor propose a possible fix and path release
20090722 Vendor released VtigerCRM 5.1.0 (Vulnerability fixed)
20090818 Advisory released

IX. CREDIT

Giovanni "evilaliv3" Pellerano, Antonio "s4tan" Parata and Francesco
"ascii" Ongaro are credited with the discovery of this vulnerability.

Giovanni "evilaliv3" Pellerano
web site: http://www.ush.it/, http://www.evilaliv3.org/
mail: evilaliv3 AT ush DOT it

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

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

X. LEGAL NOTICES

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

```
$ext_pos = strrpos($binFile, ".");
$ext = substr($binFile, $ext_pos + 1);
if (in_array(strtolower($ext), $upload_badext))
{
    $binFile .= ".txt";
}
```

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

It's known that in some circostances (for example when the PHP handler
is configured using AddType/Action/AddHandler globally, eg. not inside
an Apache's Files/FilesMatch directive) blacklisting is not enough as
files in the form of "filename.php.foo" will be mapped back to PHP
anyway (since foo is not explicitly defined in the MIME map and Apache
will try to guess the filetype by its own).

```
Beside this known issue we want to point out a less known exploitation
methodology that works on Windows hosts.

First the attacker has to find the name of the file that was uploaded
in the attachment list files. Vtiger CRM saves files in a path like:

storage/2009/July/week1/

And prepends an incremental unique number to the filename like:

133_foo.php

So, a hypothetical attacker has only to guess the prepended number. This
can be done by bruteforcing or by requesting the url:

http://127.0.0.1/vtigercrm/index.php?module=Emails&action=ListView

At this page Vtiger CRM shows the list of all the emails sent and saved,
and for every email it allows to download the attachment showing its
unique id in the link.

http://127.0.0.1/vtigercrm/index.php?module=uploads&action=downloadfile&
return_module=Emails&fileid=133&entityid=136

So, finally, the link to exploit this vulnerability should be something
like:

http://127.0.0.1/vtigercrm/storage/2009/July/week1/133.foo.php

While Vtiger CRM blocks known dangerous extensions (like .php) making
direct exploitation impossible it has to be highlighted that this simple
estension check is totally improper since it does not consider specific
filenames and behaviours of the operating systems where Vtiger CRM is
deployed.

For example on Windows OS is possible to exploit this vulnerability by
requesting an upload with the filename "foo.php.".

This string will bypass the check and since Windows does not permit
filenames ending with a dot, modifying it in a transparent way, the final
name of the file will simply be "foo.php.".

A similar result can be obtained on GNU/Linux by requesting an upload
with the filename "foo.php/."

Note that the integrated webmail feature that allows a user to write
emails and eventually save a draft of them is authenticated (a valid
user on the system is required in order to exploit this vulnerability).

B) Multiple CSRF (Cross Site Request Forgery) Vulnerabilites

Multiple CSRF vulnerabilities exist in vtiger crm version 5.0.4.
Here's a demonstrative one (an Admin user has to follow this link):

http://127.0.0.1/vtigercrm/index.php?module=Rss&action=Save&rssurl=http:
//www.ush.it/feed

The feed is added to the news feed system visible by the crm users.

Other and more dangerous CSRF vulnerabilities exist.

C) Local File Inclusion

Some LFI vulnerabilities exist in Vtiger CRM version 5.0.4.

Some examples:

1) http://127.0.0.1/vtigercrm/graph.php?module=/../[..]/../etc/passwd%00
2) http://127.0.0.1/vtigercrm/index.php?module=Accounts&action=Import&pa
renttab=Support&step=/../[..]/../etc/passwd%00

Add as many "../" instead of the "[..]" placeholder as needed.

The first one does not need a valid user account, the second one is
authenticated.

Other modules are vulnerable to LFI, for example those who include
"Import/index.php" where the vulnerability resides:

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

grep "Import/index.php" * -R

modules/Accounts/Import.php: include('modules/Import/index.php');
modules/Contacts/Import.php: include('modules/Import/index.php');
modules/HelpDesk/Import.php: include('modules/Import/index.php');
modules/Leads/Import.php: include('modules/Import/index.php');
modules/Potentials/Import.php: include('modules/Import/index.php');
modules/Products/Import.php: include('modules/Import/index.php');
modules/Vendors/Import.php: include('modules/Import/index.php');

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

A third LFI vulnerability has been found in "CommonAjax.php", both
"module" and "file" parameters are vulnerable.

http://127.0.0.1/vtigercrm/include/Ajax/CommonAjax.php?module=Email&file=bar

Will lead to a call like "require_once(modules/Email/bar.php)".

If direct access to "CommonAjax.php" has been forbidden other entry
points can be used:

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

grep "Ajax/CommonAjax.php" * -R
modules/Campaigns/CampaignsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/SalesOrder/SalesOrderAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/System/SystemAjax.php: require_once('include/Ajax/CommonAjax.php');
modules/Products/ProductsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/uploads/uploadsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Dashboard/DashboardAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Potentials/PotentialsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Notes/NotesAjax.php: require_once('include/Ajax/CommonAjax.php');
modules/Faq/FaqAjax.php: require_once('include/Ajax/CommonAjax.php');
modules/Quotes/QuotesAjax.php: require_once('include/Ajax/CommonAjax.php');
modules/Utilities/UtilitiesAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Calendar/ActivityAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Calendar/CalendarAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/PurchaseOrder/PurchaseOrderAjax.php:
require_once('include/Ajax/CommonAjax.php');
```

```
modules/HelpDesk/HelpDeskAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Invoice/InvoiceAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Accounts/AccountsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Reports/ReportsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Contacts/ContactsAjax.php:
require_once('include/Ajax/CommonAjax.php');
modules/Portal/PortalAjax.php: require_once('include/Ajax/CommonAjax.php');

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

To use one of these files as gateway for the previous vulnerability
issue a request like the following:

http://127.0.0.1/vtigercrm/?module=Invoice&action=InvoiceAjax&file=bar

Where "Invoice" and "InvoiceAjax" are values from the presented list.

This LFI vulnerability is not exploitable if you have applied a separate
patch available at the following url:

https://sourceforge.net/projects/vtigercrm/files/vtiger%20CRM%205.0.4%20
Latest%20Stable/VtigerCRM504_Security_Patch.zip

We question ourself about the usefulness of such patch without a proper
release. Probably little or no Vtiger CRM customers have applied such
patch.

D) Cross Side Scripting vulnerabilites

Some XSS vulnerabilities exist in Vtiger CRM version 5.0.4.

For example:

http://127.0.0.1/vtigercrm/phprint.php?module=Activities&action=--%3E%3C
script%3Ealert(%22ush.it%22);%3C/script%3E%3C!--

Or:

http://127.0.0.1/vtigercrm/index.php?action=UnifiedSearch&module=Home&pa
renttab=My+Home+Page&query_string=%27%22%3E%3Cscript%3Ealert(123)%3C/scr
ipt%3E

IV. DETECTION

Vtiger CRM 5.0.4 and possibly earlier versions are vulnerable.

V. WORKAROUND

Upgrade to latest version 5.1.0.

VI. VENDOR RESPONSE

"Our team reviewed the issues reported against current development build
(version 5.1.0) and seem to have
addressed many of them already. In this version we have made several
improvements to performance and
closed loop holes reported on 5.0.4 with lot more features.

Please let me know if you need further clarification.
Thank you for your support once again."

VII. CVE INFORMATION

No CVE at this time.

VIII. DISCLOSURE TIMELINE

20090620 Bug discovered
20090706 First vendor contact
20090706 Vendor Response
20090706 Vendor Confirm the vulnerability
20090713 Vendor propose a possible fix and path release
20090722 Vendor released VtigerCRM 5.1.0 (Vulnerability fixed)
20090818 Advisory released

IX. CREDIT

Giovanni "evilaliv3" Pellerano, Antonio "s4tan" Parata and Francesco
"ascii" Ongaro are credited with the discovery of this vulnerability.

Giovanni "evilaliv3" Pellerano
web site: http://www.ush.it/, http://www.evilaliv3.org/
mail: evilaliv3 AT ush DOT it

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

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

X. LEGAL NOTICES

Copyright (c) 2009 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.

# milw0rm.com [2009-08-18]
```

Tags:

Advisory/Source: Link

Databases

Links

Sites

Solutions