Vulnerability Check

Startupnomands.com

Report on the security flaws of the website startupnomands.com prepared based on the request from Mr. Federico Baladelli, CEO Startup Nomads

1/7/2013

Table of Contents

Security Check environment 3

Source files & URL wise security vulnerabilities 3

Admin Panel 3

URL : /admin/ 3

Vulnerability: SQL Injection 4

Same Login Credentials for Admin and cPanel 4

Directory Index 5

Vulnerability: Can list files in a directory 5

Session Vulnerabilities 5

Test Codes in Production Environment 6

Vulnerability: Printing test codes and values 6

Vulnerability in Admin/accelerator_areas.php 6

Vulnerability Type: SQL Injection 6

File : Admin/accelerators_edit.php 6

File : Admin/accelerators_edit_photos.php 6

File : admin/add_vdo.php 7

File : admin/add_awimg.php 7

Vulnerability : 7

File : admin/add_galery.php 7

Vulnerability : 7

Vulnerability : 8

File : admin/add_news_letter.php~ 8

File: admin/add_portfolio.php~ 8

File: admin/add_scribles.php 8

File: admin/admin.php~ 8

File: admin/edit_awrd_image.php 9

File: admin/edit_bioimg.php 9

File: admin/edit_cms.php 9

File: admin/edit_content.php 9

File: admin/edit_gallery.php 10

File: admin/edit_gallery_image.php 10

File: admin/edit_home.php 10
File: admin/edit_sub.php 10
File: admin/editvideo.php 10
File: admin/export.php 10
File: admin/editvideo.php 11
Summarized Note 11
General Vulnerabilities 11
Remarks 12

Security Check environment

IP: **117.253.175.121**
OS: Ubuntu Linux OS / Windows 7
Browser: Google Chrome
Conducted by: Manu S Ajith (manu@rubykitchen.org)

Source files & URL wise security vulnerabilities

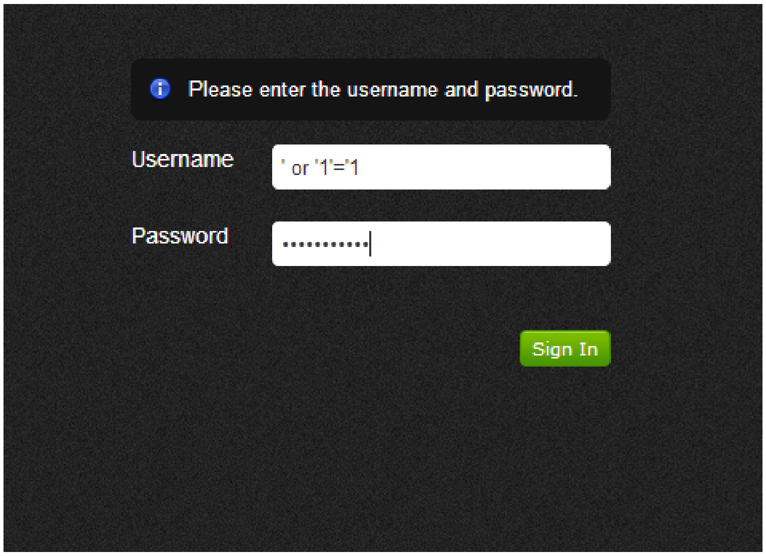
Provided below are the sections and pages of the website startup nomads where we found the vulnerabilities.

Admin Panel

URL : /admin/

Un-authorized entry can be achieved

```
Test case:
username: ' or '1'='1
Password: ' or '1'='1
```



Please enter the username and password.

Username: ' or '1'='1

Password: ••••••••••••

Vulnerability: SQL Injection

Prevention:

The username and passwords should be escaped directly from the POST variables, and not after they are hashed.
Because the when a hacker tries with password ' or 1=1-- it will be bypassed and he can easily have the admin privileges.

Using the stripslashes function of php

```
function preparePost($name) {
  if (!isset($_POST[$name]))
      return false;
  if (get_magic_quotes_gpc())
      $_POST[$name] = stripslashes($_POST[$name]);
  Return trim(strip_tags($_POST[$name]));
}
```

This should be fixed in the admin login script located at :

/public_html/admin/includes/login/login.php

Same Login Credentials for Admin and cPanel

It is not advised to have same login credentials for your admin panel, hosting cPanel and databases.
Vulnerability: If admin password can be guessed or the database gets hacked by SQL injection methods or any other ways you will be vulnerable of losing your hosting credentials too. So a black-hat hacker can easily use your server for malicious activities.

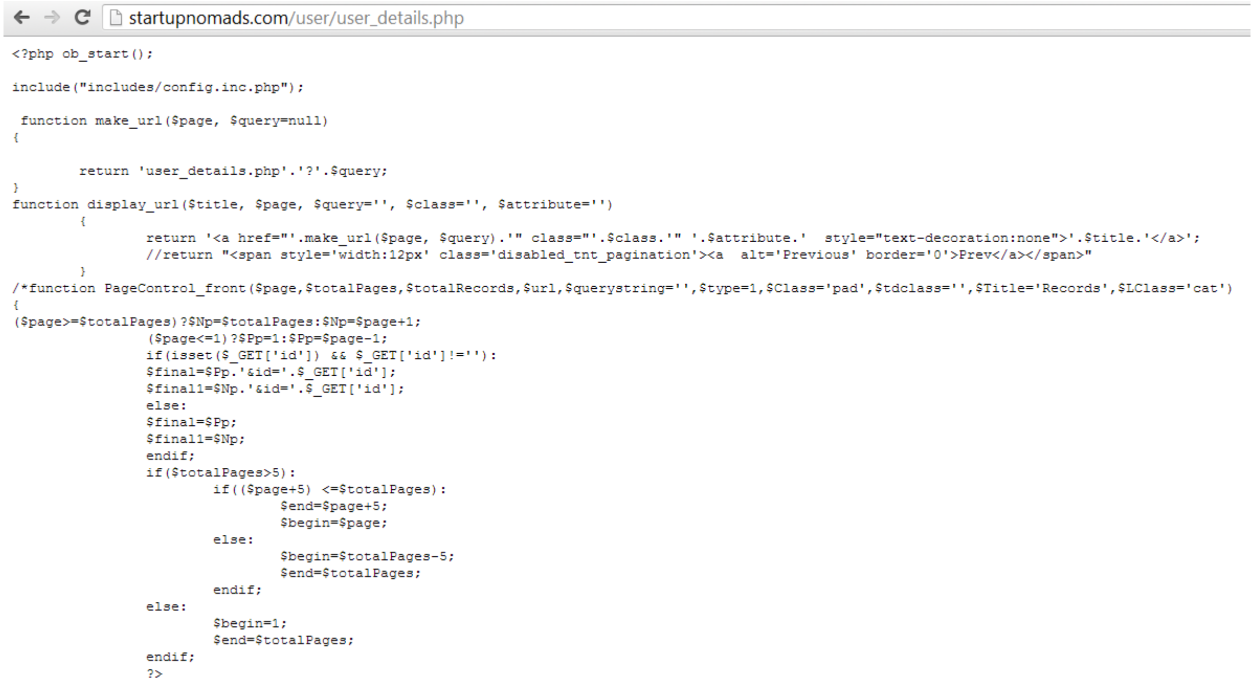
Directory Index

Vulnerability: Can list files in a directory

The directories in the site doesn't have any directory index files. So a hacker can simply navigate to a folder and can see the files in that folder.
Eg: A hacker can navigate into /pdf or /user or /includes … and get the files in that particular directory or folder. These folders will have confidential details as in the case of /pdf folder and in worst cases he can even have the list of the configuration files in case he navigates to /includes folder.

Prevention:

Directory listing should be disabled in the cPanel



```
<?php ob_start();

include("includes/config.inc.php");

 function make_url($page, $query=null)
{

        return 'user_details.php'.'?'.$query;
}
function display_url($title, $page, $query='', $class='', $attribute='')
        {
                return '<a href="'.make_url($page, $query).'" class="'.$class.'" '.$attribute.'  style="text-decoration:none">'.$title.'</a>';
                //return "<span style='width:12px' class='disabled_tnt_pagination'><a  alt='Previous' border='0'>Prev</a></span>"
        }
/*function PageControl_front($page,$totalPages,$totalRecords,$url,$querystring='',$type=1,$Class='pad',$tdclass='',$Title='Records',$LClass='cat')
{
($page>=$totalPages)?$Np=$totalPages:$Np=$page+1;
                ($page<=1)?$Pp=1:$Pp=$page-1;
                if(isset($_GET['id']) && $_GET['id']!=''):
                $final=$Pp.'&id='.$_GET['id'];
                $final1=$Np.'&id='.$_GET['id'];
                else:
                $final=$Pp;
                $final1=$Np;
                endif;
                if($totalPages>5):
                        if(($page+5) <=$totalPages):
                                $end=$page+5;
                                $begin=$page;
                        else:
                                $begin=$totalPages-5;
                                $end=$totalPages;
                        endif;
                else:
                        $begin=1;
                        $end=$totalPages;
                endif;
                ?>
```

Figure 1 – http://startupnomads/user/user_details.php

Session Vulnerabilities

The site is highly vulnerable to session vulnerabilities. Files under the admin folder are accessible to even the public. He/she may not be a logged in as admin or even as a normal user.
As long as the sessions are not validated properly there is no point in user levels and user level permissions.

Test Codes in Production Environment

Vulnerability: Printing test codes and values

Printing or displaying test codes in the production environment is not a good practice. These test codes are often used to display or test for conditions or values. If in real time production environment these conditions fails and any of the users have access to any of the details in the test code, it can be dangerous.
I saw some test codes in some files which are used to display the session variables. If in any case these test codes are somehow displayed in real time environment then a hacker can use these values for session hijacking or session spoofing.

Vulnerability in Admin/accelerator_areas.php

Vulnerability Type: SQL Injection

The post variables are not escaped for new_area, edit_area
Every data that is retrieved from the form should be escaped using mysql_real_escape_string and stripslashes.

File : Admin/accelerators_edit.php

This file meets the general security considerations of SQL-injection. The POST values are escaped.
Vulnerability :

Session management

File : Admin/accelerators_edit_photos.php

Vulnerability :

File RFI (Remote File Inclusion)

If some hacker gets into the admin panel, he/she has the privilege to upload files to your server. He/she

can upload malicious files to your server and can even use your server for DDoS-ing (Distributed Denial of Service) other servers or even can be used to send spam or bulk messages to other users.
Test case: I uploaded a file named test.php and there was no validation for .php extensions. Even though the file does not have execute permissions in this case, if that files is moved to a location having executable permissions, the hacker can execute that script.
NB: the file I have uploaded is still there, in the location public_html/user/camps/1357470972test.php
Please delete the file once you read this report.
Prevention : The file extension should be validated before being uploaded into the site. Files having extensions like .jpg, .png etc only should be allowed to be uploaded. Also hackers will try to upload images using the file extension image.php.jpg So in the code there should be validation for the same too.

File : admin/add_vdo.php

Vulnerability :

Session management and RFI
No validation for sessions. Anyone can access the file and can upload videos or any malicious scripts to the site. The user or the attacker don't even want to be an admin or even a registered user at the site.
Prevention : Use of session validation scripts.

File : admin/add_awimg.php

Vulnerability :

Session management and RFI
Same as above

File : admin/add_galery.php

Vulnerability :

Session management
Same as above

File : admin/add_images.php

Vulnerability :

Session management and RFI
Same as above

File : admin/add_news_letter.php~

Vulnerability :

Session management and RFI
Same as above and also since the file doesn't have a proper file extension the code is exposed to the public.

File: admin/add_portfolio.php~

Vulnerability:

Session management and RFI
Same as above and also since the file doesn't have a proper file extension the code is exposed to the public.

File: admin/add_scribles.php

Vulnerability:

Session management and RFI
Same as above and also since the file doesn't have a proper file extension the code is exposed to the public.

File: admin/admin.php~

Vulnerability:

Since the file doesn't have a proper file extension the code is exposed to the public.

File: admin/change_password.php

Vulnerability:

No session validation, the code is accessible to public. No SQL escaping for new passwords.

File: admin/content_page.php

Vulnerability:

No session validation, Code accessible to public.
Database connection errors are displayed publicly, thus exposing your database user

File: admin/edit_awrd_image.php

Vulnerability :

No session validation, Page can be access by public.

File: admin/edit_bioimg.php

Vulnerability:

Same as above

File: admin/edit_cms.php

Vulnerability:

No session validation, Code accessible to public.
Database connection errors are displayed publicly, thus exposing your database user

File: admin/edit_content.php

Vulnerability:

No session validation, Code accessible to public..
Can be used to post values to other pages if the fields are known to public. Hence potentially harmful.

File: admin/edit_gallery.php

Vulnerability:

No session validation. Database errors will display the database users.

File: admin/edit_gallery_image.php

Vulnerability:

No session validation, File accessible by public, Database errors will display the database users.

File: admin/edit_home.php

Vulnerability:

No session validation, File accessible by public, Database errors will display the database users.

File: admin/edit_sub.php

Vulnerability:

No session validation, File accessible by public, Database errors will display the database users.

File: admin/editvideo.php

Vulnerability:

No session validation, File accessible by public, Database errors will display the database users.

File: admin/export.php

Vulnerability:

No session validation, file accessible to public. This file exports the users database to CSV, This database will contain all your registered users login credentials, their personal and professional details and all the details that you collect on your website.
If this data becomes public then it will lead to violation of your T&C and privacy policy.

File: admin/editvideo.php

Vulnerability:

No session validation, File accessible by public, Database errors will display the database users.

All the above files don't have proper session validation. The files are accessible to the public. Also if hacker can include the database credential files in the include folder, he will be able to connect with the database and will have full access to the files and its actions.

Summarized Note

- All values from POST should be strip slashed first, and mysql_real_escape_string to prevent SQL-Injection attacks
- All the files which should be under the login should be validated for sessions.
- Check for session hijacking, Regenerate session variables to avoid session hijacking and session spoofing.
- All the values from the POST should be escaped in order to avoid Cross Site Scripting (XSS) attacks.
- All the forms should have a form id to prevent Cross Site Forgery Request(CSRF) attacks
- All the file upload forms should validate the file types and extensions in order to prevent Remote File Inclusion (RFI) and Local File Inclusion (LFI) type of attacks.
- Remove all test codes from the files which is used for debugging the code
- Proper file extension to avoid code visibility
- Disable directory listing to prevent listing of files in folders/directories

General Vulnerabilities

Since the website is hosted on a shared server it is vulnerable to hacking attempts, from other website. If a hacker can hack into the server using other sites and upload a shell script to the server and root the whole server he can deface your site too.

So it would be advisable to move to a Virtual Private Server (VPS.)

Remarks

From the analysis we noticed that the codes are actually written by two set of people, as the files under admin panel were the codes which had all the vulnerabilities. Those outside were written taking into consideration these security vulnerabilities, we recommend the person who wrote the second set of codes to look into the codes under the admin folder.