

Breakdown of Key Metrics

High Risk

Medium Risk

Low Risk

Informational

2

Issues

4

Issues

2

Issues

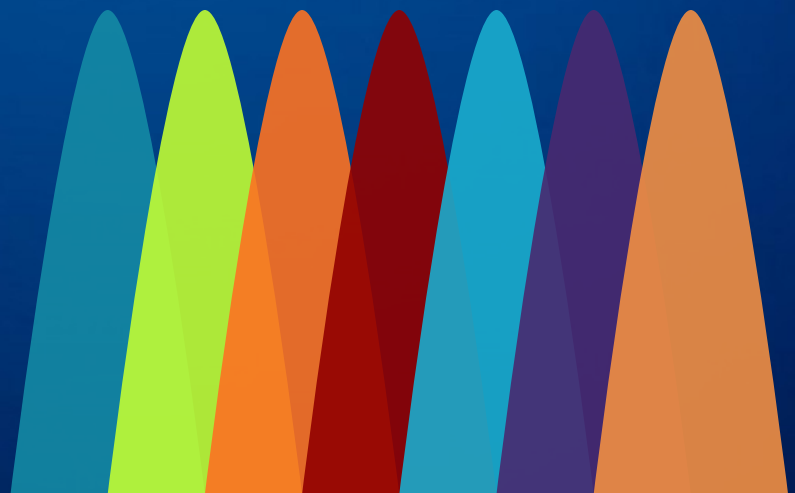
3

Issues



Severe Risk

Insufficient Input
Validation
Insufficient Webroot
Management
Insecure Configuration
Patch Management
Insufficient Patch
Management
Server Configuration
Application
Misconfiguration



50% of all high-risk issues could be resolved by fixing the "Insufficient Input Validation" issue.

Overview

About Application

This application provides different wallpapers and videos for the following functionalities.

- Wallpapers that Inspire
- Videos for Kids
- Amazing science facts
- General Knowledge
- History



Note

This report has been generated to contain issues which are on educstar.com.

Scope of Work

The following hosts were considered to be part of the scope of this engagement.

No.	URL	Description
1	educstar.com	WAP Based 1 Video Portal (1st Application)

Timelines

The project EduCStar was started on 2017-05-05 and completed on 2017-05-09.

Milestone	Start Date	End Date
-----------	------------	----------

Vulnerabilities

Title	Risk	Status
SQL Injection Vulnerabilities	High	Vulnerable
Source Code Files are Exposed and Publicly Accessible	High	Vulnerable
Directory Listing enabled on the Server	Medium	Vulnerable
PHP Outdated Version contains Several Vulnerabilities	Medium	Vulnerable
Cross-Site Scripting Vulnerabilities	Medium	Vulnerable
Apache Outdated Version Contains Several Security Vulnerabilities	Medium	Vulnerable
In-secure HTTP Methods are Enabled	Low	Vulnerable
Cookie without HTTPOnly flag set	Low	Vulnerable
Potential Clickjacking Vulnerability	Informational	Vulnerable
Information Disclosure by Verbose Error Messages	Informational	Vulnerable
Information Disclosure by Test, Old and Backup Files	Informational	Vulnerable

SQL Injection Vulnerabilities

High Risk

Vulnerable

Description

Websites often use databases at the backend to store and manage large amounts of information. The de-facto standard language for querying databases is SQL. Web applications often take user input (taken out of the HTTP request) and incorporate it in an SQL query, which is then sent to the backend database. The query results are then processed by the application and sometimes displayed to the user.

By exploiting this vulnerability, an attacker can directly pass malicious queries and inputs to the database and interpret the responses from the database. It allows an attacker to read, write, modify or delete information stored within the database along with sometimes gaining system level access to the underlying operating system.

In the instances below, the affected parameters were passing user input directly to the back-end database without proper validation. Because of this, it is possible to gain complete access to the database. On successful exploitation of the vulnerability, an attacker would have access to Read, Write and Modify any data stored within the database.

Database Compromise

Impact

Insufficient Input Validation

Cause

Developer

Responsibility

Medium

Difficulty

Critical / CVSS Base Score: **9.6**

(AV:N/AC:L/Au:N/C:C/I:C/A:P)

Proof-Of-Concept

Step 1 : While testing the application we were able to inject malicious query and able to enumerate database name on the server.

```

[15:19:01] [INFO] retrieved: content_educstar
[15:21:35] [INFO] retrieved: content_fitzox
[15:23:57] [INFO] retrieved: content_gamezuff
[15:26:35] [INFO] retrieved: content_mtnpinacle
[15:29:44] [INFO] retrieved: content_right2study
[15:33:19] [INFO] retrieved: content_six4fix
[15:35:48] [INFO] retrieved: content_stylebox
[15:38:29] [INFO] retrieved: content_walobby
[15:40:53] [INFO] retrieved: content_wapheros
[15:43:18] [INFO] retrieved: login_table
[15:46:24] [INFO] retrieved: tbl_admin
[15:48:50] [INFO] retrieved: tbl_category

```

```

Database: allpanel
[16 tables]

```

```

+-----+
| check_mail |
| content_Afrimobitv |
| content_Tv9jia |
| content_demo |
| content_educstar |
| content_fitzox |
| content_gamezuff |
| content_mtnpinacle |
| content_right2study |
| content_six4fix |
| content_stylebox |
| content_walobby |
| content_wapheros |
| login_table |
| tbl_admin |
| tbl_category |
+-----+

```

Step 2 : After retrieving the database name we then enumerated the tables present in the database.

```

Type: stacked queries
Title: MySQL > 5.0.11 stacked queries (SELECT - comment)
Payload: username=admin';(SELECT * FROM (SELECT(SLEEP(5)))
min&panel_id=Educstar_Panel/index.php&submit=

```

```

[16:35:48] [INFO] the back-end DBMS is MySQL
web server operating system: Linux CentOS 6.5
web application technology: PHP 5.3.3, Apache 2.2.15
back-end DBMS: MySQL 5.0.11
[16:35:48] [INFO] fetching columns for table 'tbl_admin' in dat
[16:35:48] [WARNING] running in a single-thread mode. Please co
option '--threads' for faster data retrieval
[16:35:48] [INFO] retrieved:
[16:35:53] [WARNING] reflective value(s) found and filtering on
27
[16:36:07] [INFO] retrieved: id
[16:36:41] [INFO] retrieved: int(10)
[16:38:32] [INFO] retrieved: user_type
[16:41:09] [INFO] retrieved: varchar(50)
[16:43:56] [INFO] retrieved: username
[16:45:55] [INFO] retrieved: varchar(150)
[16:49:05] [INFO] retrieved: password
[16:51:27] [INFO] retrieved: varchar(50)
[16:54:21] [INFO] retrieved: user_name
[16:56:45] [INFO] retrieved: varchar(250)
[16:59:43] [INFO] retrieved: user_

```

Step 3 : As can be seen, with SQL injection exploitation we were able find out the columns of the table content_educstar.

```

Database: allpanel
Table: content_educstar
[6 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| category | varchar(255) |
| content | varchar(255) |
| id | int(11) |
| parent | varchar(255) |
| update_time | datetime |
| video | varchar(255) |
+-----+-----+

[16:21:17] [INFO] fetched data logged to text file
p\output\213.136.71.172'

[*] shutting down at 16:21:17

```

```

Database: allpanel
Table: content_educstar
[1 entry]
+-----+-----+-----+-----+-----+-----+
| id | video | parent | content | category | update_time |
+-----+-----+-----+-----+-----+-----+
| 1 | 1 | Videos | http://213.136.71.172/content/Galactic_Portal/Kids/20-01-2017/thumbnails/H - N phonics.jpg | Kids Videos | 2017-04-05 17:14:38 |
+-----+-----+-----+-----+-----+-----+

```

Recommendations

SQL injection and blind SQL injection:

There are four possible ways to protect your web application against SQL injection attacks.

1. Store Procedures

Use a stored procedure rather than dynamically built SQL query string. The way parameters are passed to SQL Server stored procedures, prevents the use of apostrophes and hyphens. After the storing of the commands is done, the tasks can be performed or executed continuously, without being repeatedly sent to the server. This also helps in decreasing the traffic in the networks and also reduces the CPU load.

Why use a stored procedure?

Develop the functionality once and all the applications can call the same commands.

Network Traffic reduced to a greater extent.

Centralization of all commands made possible, which is helpful for various applications that repeatedly call the same set of complicated commands.

Runs on any kind of environment.

Create a Stored Procedure in the Database example for PHP

```

/**
 * @author Security Brigade InfoSec Private Limited
 * @project Automated Application Security Audit of EduCStar for Coolbox Innovation Studio Private
Limited
 */

DELIMITER $$
// Check if stored procedure exists, if yes delete it and create new one.
DROP PROCEDURE IF EXISTS `UName`.`get_user`$$
CREATE PROCEDURE `UName`.`get_user`
(

// Pass the value for userId to the stored procedure
IN userId INT,

// Return the value after the stored procedure has executed
OUT firstName VARCHAR(100),
OUT lastName VARCHAR(100)
)
BEGIN
SELECT first_name, last_name
INTO firstName, lastName
FROM users
WHERE users_id = userId;
END $$
DELIMITER ;

```

2. Parameterized Queries

A parameterized query is a query in which placeholders are used for parameters and the parameter values are supplied at execution time. The placeholders are typically type-specific (for example, int for integer data and text for strings) which allows the database to interpret the data strictly. For instance, a text placeholder is always interpreted as a literal, avoiding exploits such as the query stacking SQL injection. A mismatch between a placeholder's type and its incoming datum causes execution errors, adding further validation to the query.

This is the absolute best method to prevent any SQL injection. Although complex nature of the code makes it bit difficult to implement.

3. Input Validation

You can add input validation to Web Forms pages by using validation controls. Validation controls provide an easy-to-use mechanism for all common types of standard validation - for example, testing for valid dates or values within a range - plus ways to provide custom-written validation. In addition, validation controls allow you to completely customize how error information is displayed to the user. Validation controls can be used with any controls that are processed in a Web Forms page's class file, including both HTML and Web server controls.

In the below example the value for age is validated using an int function. This would prevent user from passing any string character to the database. This is not a full-proof method to prevent SQL injection but it will make attacker's job a bit harder.



Input Validation example for PHP

```
/**
 * @author Security Brigade InfoSec Private Limited
 * @project Automated Application Security Audit of EduCStar for Coolbox Innovation Studio Private
Limited
 */

<?php
$name = htmlspecialchars($_POST['name'], ENT_QUOTES);
echo $name;
?>
```

Instances

educstar.com

URL	Parameters	Status	POC
msn.php	Service_id=	Vulnerable	
getpost.php	parent	Vulnerable	

Reference Documents

GENERAL

- [SQL Injection - WASC](#)
- [CAPEC-66: SQL Injection](#)
- [SQL Injection - OWASP](#)

Source Code Files are Exposed and Publicly Accessible

High Risk

Vulnerable

Description

Source code disclosure vulnerability allows an attacker to view / download the source code files of the web application from the webserver. These files can reveal sensitive information related to specific functionality of the website, how different web-pages react to various parameter values given to it etc. An attacker can use this information to carry out targeted attacks on the website which may lead to website compromise.

The files and directories listed above contain source-code files that are publicly accessible.

An attacker may download these source-code files and carry out intellectual property theft.

In-addition, these files may contain sensitive details about the inner workings of the website and disclose additional vulnerabilities.

Website Compromise

Impact

Insufficient Webroot Management

Cause

System Administrator

Responsibility

Easy

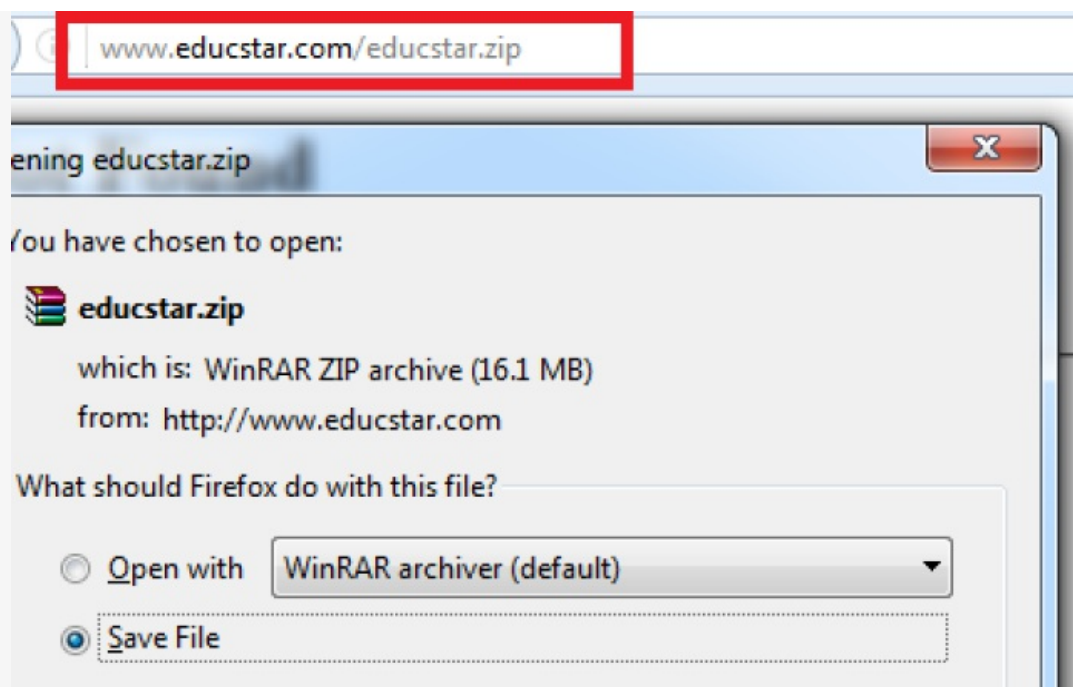
Difficulty

Critical / CVSS Base Score: **7.8**

(AV:N/AC:L/Au:N/C:C/I:N/A:N)

Proof-Of-Concept

As can be seen in the below screen shots, we were able to get the source code of the application pages that were showing the sensible data.



```
elseif($video==2)
{
    $p=$row3['content'];
    $fname=basename($p);
    $exp_fname=explode('.', $fname);
    $ex=explode('/', $p);
    $cou="-".count($ex);
    $abc=array_splice($ex, $cou, -2);
    $imp=implode('/', $abc);
    $file_to_download = $imp."/". $exp_fname[0];
    $path = $p;

    ?>
    <a href="check.php?path=<?php echo $file_to_download.".apk"?>">
    <!--<a href="checkdownload.php?path=<?php //echo $path;?>"-->
        
    </a>
    <p id="content_name"><?php echo substr($exp_fname[0],0,30)."..."?></p>
    <?php
}
```

```

<?php
include_once('dbconfig.php' );
include('function/gbFunction.php');
$DBCClass=new GlobalClass();

function generate_random_password($length = 10)
{
    $alphabets = range('A','Z');
    $numbers = range('0','9');
    $additional_characters = array('!','@','#','$','%','&',
    $final_array = array_merge($alphabets,$numbers,$addi
    $password = '';

    while($length-->0) {
        $key = array_rand($final_array);
        $password .= $final_array[$key];
    }
}

```

```

<?php
date_default_timezone_set('Asia/Calcutta');
$host = 'localhost';
$username = 'root';
$password = '';
$dbname = 'voda_educstar';
?>

```

Recommendations

There are several different steps that can be taken to ensure that these vulnerabilities are removed:

1. Implement strong production and development processes to prevent unapproved files from reaching a production environment.
2. Blacklist archive extensions such as zip, rar, tar, gz etc. that may contain source-code files.
3. Carry out regular audits of the webroot and remove any unnecessary files or directories.

Instances

URL	Parameters	Status	POC
educstar.zip	–	Vulnerable	Ø

Reference Documents

PHP

- [Exposed Source Code](#)

Directory Listing enabled on the Server

Medium Risk

Vulnerable

Description

A web directory is a listing of websites organized in a hierarchy or interconnected list of categories. If Directory Listing is enabled on the web-server, an attacker can view a list of all files from this directory, possibly exposing sensitive information.

The web server is configured to display the list of all files contained in the above directory.

An attacker can use this vulnerability to view and understand the functionality of the website and use it to conduct targeted attacks on the website.

Information Disclosure

Impact

Insecure Configuration

Cause

System Administrator

Responsibility

Easy

Difficulty

Critical / CVSS Base Score: **6.4**




(AV:N/AC:L/Au:N/C:P/I:P/A:N)

Proof-Of-Concept














As can be seen in the below screen shots the directory listings are enabled publicly on the server.



Index of /includes

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 css/	03-Apr-2017 14:04	-	
 dropdown.php	24-Mar-2017 18:41	1.8K	
 footer.php	30-Mar-2017 17:13	922	
 header.php	05-Apr-2017 17:31	8.7K	
 header_5Apr17.php	03-Apr-2017 10:39	8.7K	
 home-icon.png	30-Mar-2017 15:04	4.3K	
 js/	03-Apr-2017 14:04	-	

Apache/2.2.15 (CentOS) Server at www.educstar.com Port 80

  www.educstar.com/icons/			
Index of /icons			
Name	Last modified	Size	Description
 Parent Directory		-	
 a.gif	21-Nov-2004 01:46	246	
 a.png	26-Nov-2008 12:06	306	
 alert.black.gif	21-Nov-2004 01:46	242	
 alert.black.png	26-Nov-2008 12:06	293	
 alert.red.gif	21-Nov-2004 01:46	247	
 alert.red.png	26-Nov-2008 12:06	314	
 apache_pb.gif	21-Nov-2004 01:46	2.3K	
 apache_pb.png	26-Nov-2008 12:06	2.0K	
 apache_pb2.gif	26-Nov-2008 12:06	1.8K	
 apache_pb2.png	26-Nov-2008 12:06	1.5K	

Recommendations

It's advisable to disable Directory Listing on the webserver

Disable Directory List example for Apache










```
/**
 * @author Security Brigade InfoSec Private Limited
 * @project Automated Application Security Audit of EduCStar for Coolbox Innovation Studio Private
Limited
 */
```

To prevent directory listings, create an `.htaccess` file following the main instructions [and](#) guidance which includes the following text:

```
IndexIgnore *
```

Instances

educstar.com

URL	Parameters	Status	POC
logs/	-	Vulnerable	
CG/	-	Vulnerable	
function/	-	Vulnerable	
includes/	-	Vulnerable	
image/	-	Vulnerable	
icons/	-	Vulnerable	
content/	-	Vulnerable	
logs/partnernames/partners_2017-04-03.txt	—	Vulnerable	
logs/msn_2017-04-03.txt	—	Vulnerable	

Reference Documents

APACHE

- [Disable Directory Listing](#)

PHP Outdated Version contains Several Vulnerabilities

Medium Risk

Vulnerable

Description

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. The version of PHP being used has been identified as being outdated and containing several security vulnerabilities.

The version of PHP being used has been identified as being outdated and containing several security vulnerabilities.

Here are some of the know vulnerabilities that is present in the current version 5.3.3:

- CVE-2016-7478 - It allows remote attackers to cause a denial of service (infinite loop via a crafted Exception object in serialized data).
- CVE-2014-9427 119 - It allows remote attackers to obtain sensitive information from php-cgi process memory by leveraging the ability to upload a .php file or (2) trigger unexpected code execution if a valid PHP script is present in memory locations adjacent to the mapping.
- CVE-2013-6420 119 - It allows remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via a crafted certificate that is not properly handled by the openssl_x509_parse function.

System Compromise
Impact

Patch Management
Cause

System Administrator
Responsibility

Medium
Difficulty

Critical / CVSS Base Score: **6.4**
(AV:N/AC:L/Au:N/C:P/I:P/A:N)

Proof-Of-Concept

As can be seen in the below screenshot, we found that the PHP version being used is outdated.


```
HTTP/1.1 200 OK
Date: Tue, 09 May 2017 11:11:21 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Set-Cookie: PHPSESSID=k6i2fbiene8h19cj1nh9kkgh84; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 37265
```

Recommendations

Upgrade to the latest version of PHP from the following url:
<http://php.net/downloads.php>

Instances

educstar.com

URL	Parameters	Status	POC
/	—	Vulnerable	

Reference Documents

PHP

- [PHP Security Vulnerabilities](#)
- [Download](#)

Cross-Site Scripting Vulnerabilities

Medium Risk

Vulnerable

Description

Websites often accept user input for the application to display on the screen. If the application is not careful enough with its treatment of user (attacker) input, it is possible for an attacker to inject malicious data, which when displayed on the screen can execute HTML or JavaScript code in the user's browser.

This vulnerability allows an attacker to either permanently or temporarily inject client-side code into the target website. This code executes when the page is loaded by the victim and the client-side code may carry out activities such as: stealing cookies/sessions, modifying the page contents, logging key strokes, etc.

User Account Compromise

Impact

Insufficient Input Validation

Cause

Developer

Responsibility

Medium

Difficulty

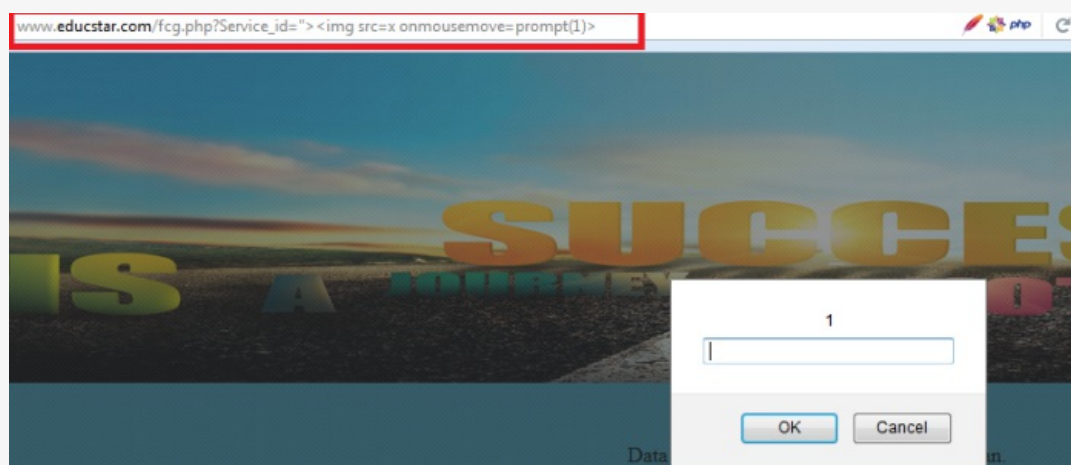
Critical / CVSS Base Score: **5.7**
(AV:N/AC:M/Au:N/C:P/I:P/A:N)

Proof-Of-Concept

Step 1 : We have intercepted the request and injected the malicious script in the URL as shown in the below screen shot.

```
GET /fcg.php?Service_id="><img src=x onmouseover=prompt(1)>
HTTP/1.1
Host: www.educstar.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:53.0) Gecko/20100101
Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Cookie: PHPSESSID=dnldq1bdr86o78ko5o7cg3srd5
Connection: close
Upgrade-Insecure-Requests: 1
```

Step 2 : As can be seen, the injected malicious script got executed successfully.



Recommendations

In-order to prevent Cross-Site Scripting issues, you can add input validation to Web Forms pages by using validation controls. Validation controls provide an easy-to-use mechanism for all common types of standard validation - for example, testing for valid dates or values within a range - plus ways to provide custom-written validation.

In addition, validation controls allow you to completely customize how error information is displayed to the user. Validation controls can be used with any controls that are processed in a Web Forms page's class file, including both HTML and Web server controls.


Input Validation example for PHP

```
/**
 * @author Security Brigade InfoSec Private Limited
 * @project Automated Application Security Audit of EduCStar for Coolbox Innovation Studio Private
 * Limited
 */

<?php
$name = htmlspecialchars($_POST['name'], ENT_QUOTES);
echo $name;
```

Instances

educstar.com

URL	Parameters	Status	POC
fcg.php	serviceid	Vulnerable	

Reference Documents

GENERAL

- [CERT Advisory CA-2000-02 Malicious HTML Tags Embedded in Client Web Requests](#)
- [The Cross-Site Scripting \(XSS\) FAQ](#)
- [Cross Site Scripting Info](#)
- [Cross Site Scripting Explained](#)

Apache Outdated Version Contains Several Security Vulnerabilities

Medium Risk

Vulnerable

Description

Apache is an open-source web server platform, which guarantees the online availability of the majority of the websites active today. The server is aimed at serving a great deal of widely popular modern web platforms.

Latest Stable Version:2.4.25

Here are some of the know vulnerabilities that is present in the current version (Apache httpd 2.2.15) :

- CVE-2011-3192 - It allows remote attackers to cause a denial of service (memory and CPU consumption) via a Range header that expresses multiple overlapping ranges.
- CVE-2014-031 - It allows remote attackers to cause denial of service (process hang) via a request to a CGI script that does not read from its stdin file descriptor.
- CVE-2014-0098 - It allows remote attackers to cause a denial of service (segmentation fault and daemon crash) via a crafted cookie that is not properly handled during truncation.

Network Compromise

Impact

Insufficient Patch Management

Cause

System Administrator

Responsibility

Easy

Difficulty

Critical / CVSS Base Score: **5.6**

(AV:L/AC:L/Au:N/C:C/I:N/A:P)

Proof-Of-Concept

As can be seen, It was found that the Apache being used is outdated




Recommendations

Upgrade the Apache httpd to the latest stable version .

Instances

educstar.com

URL	Parameters	Status	POC
/	—	Vulnerable	

Reference Documents

APACHE

- [Download Latest Version](#)

In-secure HTTP Methods are Enabled

Low Risk

Vulnerable

Description

HTTP Methods such as TRACK, TRACE, DEBUG, PUT, DELETE, OPTIONS are intended for debugging or testing purposes. Production environments that allow these HTTP methods can be vulnerable to a range of attacks that are facilitated by these HTTP methods.

The following HTTP method is enabled on the server:

- TRACE - TRACE allows the client to see what is being received at the other end of the request chain and use that data for testing or diagnostic information

Information Disclosure
Impact

Server Configuration
Cause

System Administrator
Responsibility

Easy
Difficulty

Critical / CVSS Base Score: **2.6**
(AV:N/AC:H/Au:N/C:P/I:N/A:N)

The above method would allow an attacker to upload malicious scripts onto the server as well as delete files from the server.

Proof-Of-Concept

As can be seen, we were able to find the HTTP TRACE method enabled on the server.


```
HTTP/1.1 200 OK
Date: Tue, 09 May 2017 11:35:44 GMT
Server: Apache/2.2.15 (CentOS)
Connection: close
Content-Type: message/http
Content-Length: 83
```

```
TRACE / HTTP/1.1
```

```
Host: www.educstar.com
Cookie: zx3gtc9wf9
Connection: close
```

```
TRACE / HTTP/1.1
```

```
Host: www.educstar.com
Cookie: zx3gtc9wf9
Connection: close
```

Recommendations

The changes can be made as follows:

Disable HTTP Methods example for APACHE

```
/**
 * @author Security Brigade InfoSec Private Limited
 * @project Automated Application Security Audit of EduCStar for Coolbox Innovation Studio Private
 * Limited
 */
```

IHS and Apache are configured to disable normal **TRACE** request processing so that the request fails with **403** (forbidden) and any **private** information sent in the **TRACE** request does not appear in the response. The way to disable normal **TRACE** request processing is to add several `mod_rewrite` directives to the web server configuration file, at main scope as well as in every `<VirtualHost>` container. Here is an example:

```
...
# disable TRACE in the main scope of httpd.conf
RewriteEngine On
```

```
RewriteCond %{REQUEST_METHOD}^TRACE
RewriteRule .* - [F]
```

...

```
<VirtualHost www.example.com>
```

...

```
# disable TRACE in the www.example.com virtual host
```

```
RewriteEngine On
```

```
RewriteCond %{REQUEST_METHOD}^TRACE
```


```
RewriteRule .* - [F]
```

```
</VirtualHost>
```

mod_rewrite must be active for these directives to be accepted. If mod_rewrite is not already active in your configuration

Instances

educstar.com

URL	Parameters	Status	POC
/	—	Vulnerable	

Reference Documents

APACHE

- [DISABLE TRACE AND TRACK METHODS](#)

Cookie without HTTPOnly flag set

Low Risk

Vulnerable

Description

HttpOnly is an additional flag included in a Set-Cookie HTTP response header. If the HttpOnly flag is included in the HTTP response header, the cookie cannot be accessed through client side script. As a result, even if a cross-site scripting (XSS) flaw exists, and a user accidentally accesses a link that exploits this flaw, the browser will not reveal the cookie to a third party.

The cookie value issued by the application does not have the HttpOnly flag set and hence an attacker may use it as leverage with Cross-Site Scripting to gain access sensitive parameter in the cookie value.

User Account Compromise

Impact

Application Misconfiguration

Cause

Developer

Responsibility

Medium

Difficulty

Critical / CVSS Base Score: **2.6**
(AV:N/AC:H/Au:N/C:P/I:N/A:N)

Proof-Of-Concept

As can be seen in the below screen shot, the response is showing cookie without HTTP flag only set.

```
HTTP/1.1 200 OK
Date: Tue, 09 May 2017 11:11:21 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Set-Cookie: PHPSESSID=k6i2fbiene8h19cjlnh9kkgh84; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 37265
```

Recommendations

set HTTPOnly example for APACHE

```
/**
 * @author Security Brigade InfoSec Private Limited
 * @project Automated Application Security Audit of EduCStar for Coolbox Innovation Studio Private Limited
 */
```

The **[CO]**, or **[cookie]** flag, allows you to set a cookie when a particular RewriteRule matches. The argument consists of three required fields and four optional fields.

The full syntax for the flag, including all attributes, is as follows:

[CO=NAME:VALUE:DOMAIN:lifetime:path:secure:httponly]

You must declare a name, a value, and a domain for the cookie to be set.


set in php.ini example for PHP

```
/**
 * @author Security Brigade InfoSec Private Limited
 * @project Automated Application Security Audit of EduCStar for Coolbox Innovation Studio Private Limited
 */
```

`session.cookie_httponly = True`

Instances

educstar.com

URL	Parameters	Status	POC
/	—	Vulnerable	

Reference Documents

GENERAL

- HTTP cookie
- HttpOnly

Potential Clickjacking Vulnerability

Informational Risk

Vulnerable

Description

Click jacking is a client side vulnerability which mainly occur when the attacker is able to frame the website content mostly forms and put some transparent layer over the page in order to trick the user to click on buttons that are not intended by the victim.

For example:- If there is delete profile button on the page,the attacker could easily frame the page and put a transparent layer over the page saying win lottery,the victim will click on the button with the intention of wining the lottery but he/she will lose his account.


Source Code Compromise Impact
Insecure Configuration Cause
System Administrator Responsibility
Easy Difficulty
Critical / CVSS Base Score: 0 (AV:L/AC:H/Au:M/C:N/I:N/A:N)

It was observed that the X-Frame Option header is not present in the response bodies which helps preventing from Click Jacking vulnerability.Also appropriate frame bursting javascript should be implemented in order to overcome the cases where some browsers does not support X-Frame-Options header.

Proof-Of-Concept

While testing we found that the X-frame-Options was missing from the response header.

```
Raw Headers Hex HTML Render
HTTP/1.1 200 OK
Date: Sat, 06 May 2017 07:19:57 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 4506
Connection: close
Content-Type: text/html; charset=UTF-8
```




No X-Frame Options

Recommendations

To effectively prevent framing attacks, the application should return a response header with the name X-Frame-Options and the value DENY to prevent framing altogether, or the value SAMEORIGIN to allow framing only by pages on the same origin as the response itself.

Instances

educstar.com

URL	Parameters	Status	POC
/	*	Vulnerable	

Reference Documents

GENERAL

- [Enable anti-clickjacking X-Frame-Options header](#)

Information Disclosure by Verbose Error Messages

Informational Risk

Vulnerable

Description

The website contains files and directories that result in verbose error messages that disclose information about the internal operation and functioning of the application. Information related to the website such as the framework, languages and other functions are revealed in a verbose error messages. This information provides an attacker a detailed insight about the website.

An attacker is able to trigger verbose error messages on various parts of the server and application. As a result an attacker may be able to view sensitive internal information that may be used to carry out additional attacks.

Information Disclosure Impact
Server Configuration Cause
System Administrator Responsibility
Easy Difficulty
Critical / CVSS Base Score: 0 (AV:N/AC:L/Au:N/C:N/I:N/A:N)

Proof-Of-Concept

While testing the application we found that the application was displaying verbose error message.




Recommendations

There are two key aspects to tackling this vulnerability:

1. Writing code with managed errors
2. Disabling error messages in server configuration

Instances

educstar.com

URL	Parameters	Status	POC
fcg.php	—	Vulnerable	

Reference Documents

APACHE

- [Replacing Verbose Errors with Custom Messages](#)

Information Disclosure by Test, Old and Backup Files

Informational Risk

Vulnerable

Description

The webserver contains files and directories that have been created for testing or development purposes. These files may disclose information about the website and its functioning. The more information an attacker learns, the easier it becomes for him to compromise the system.

The webserver contains files and directories that have been identified as test or debug resources. These files generally contain internal information and have not gone through a thorough release cycle. As a result, they may disclose sensitive information or be vulnerable to attacks and compromise the security of the server.

Information Disclosure

Impact

Insufficient Webroot Management

Cause

System Administrator

Responsibility

Easy

Difficulty

Critical / CVSS Base Score: **0**
(AV:N/AC:L/Au:N/C:N/I:N/A:N)

Proof-Of-Concept

As can be seen in the below screen shots, we were able to access the log files publicly enabled on the server.

Index of /logs/logs

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 callback 2017-04-12.txt/	12-Apr-2017 10:30	-	
 callback 2017-04-19.txt/	19-Apr-2017 13:51	-	
 callback 2017-04-20.txt/	20-Apr-2017 10:41	-	
 callback 2017-04-21.txt/	21-Apr-2017 10:56	-	
 callback 2017-05-08.txt/	08-May-2017 11:31	-	
 callback 2017-05-09.txt/	09-May-2017 13:03	-	
 msn 2017-04-03.txt/	03-Apr-2017 16:24	-	
 msn 2017-04-06.txt/	06-Apr-2017 05:14	-	

Index of /logs

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 CallSubUnsub/	03-Apr-2017 12:16	-	
 Calling00/	03-Apr-2017 12:15	-	
 Calling11/	03-Apr-2017 12:16	-	
 Headers/	09-May-2017 01:07	-	
 act case/	09-May-2017 13:03	-	
 callbackresponse/	03-Apr-2017 12:15	-	
 checkHEdownload-manual/	09-May-2017 13:03	-	
 hitscapture/	09-May-2017 01:07	-	
 hitscapturetosagarsir/	09-May-2017 01:07	-	
 logs/	09-May-2017 17:19	-	


Recommendations

There are several different steps that can be taken to ensure that these vulnerabilities are removed:

- 1. Implement strong production and development processes to prevent unapproved files from reaching a production environment.
- 2. Carry out regular audits of the webroot and remove any unnecessary files or directories.

Instances

educstar.com

URL	Parameters	Status	POC
/	logs	Vulnerable	

Reference Documents

GENERAL

- [SSL Cache](#)

Recommendation

Recommendation Summary

1. Implement input validation
2. Implement strong production and development processes to prevent unapproved files from reaching a production environment
3. Disable Directory Listing
4. Upgrade to the latest version of PHP
5. Implement input validation effectively.
6. Upgrade the Apache httpd to the latest stable version.
7. Disable in-secure HTTP methods
8. set HTTPOnly flag
9. It is recommended to implement X-Frame-Options
10. Writing code with managed errors

Long-Term Action Plan

Security Brigade Infosec Private Limited recommends the following Action Plan to enhance the long-term security posture at Coolbox Innovation Studio Private Limited.

Actionable Items	Priority
Comprehensive Web Application Penetration Testing	High
Application Malware Scan	High
Penetration Testing Service	Medium
Source Code Security Review	Low