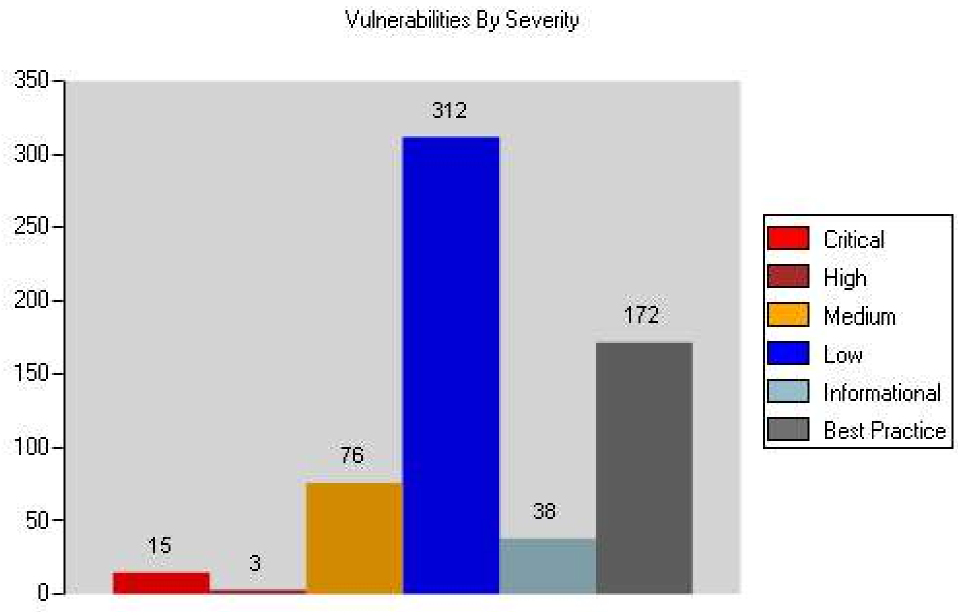
HP Fortify WebInspect

Vulnerability (Legacy)

Web Application Assessment Report

Scan Name:	SIMBPPK frontend - P2 - Santi		
Policy:	Standard	**Crawl Sessions:**	1059
Scan Date:	2/18/2015 4:52:28 PM	**Vulnerabilities:**	406
Scan Version:	10.30.507.10	**Scan Duration:**	3 hours : 16 minutes
Scan Type:	Site	**Client:**	FF

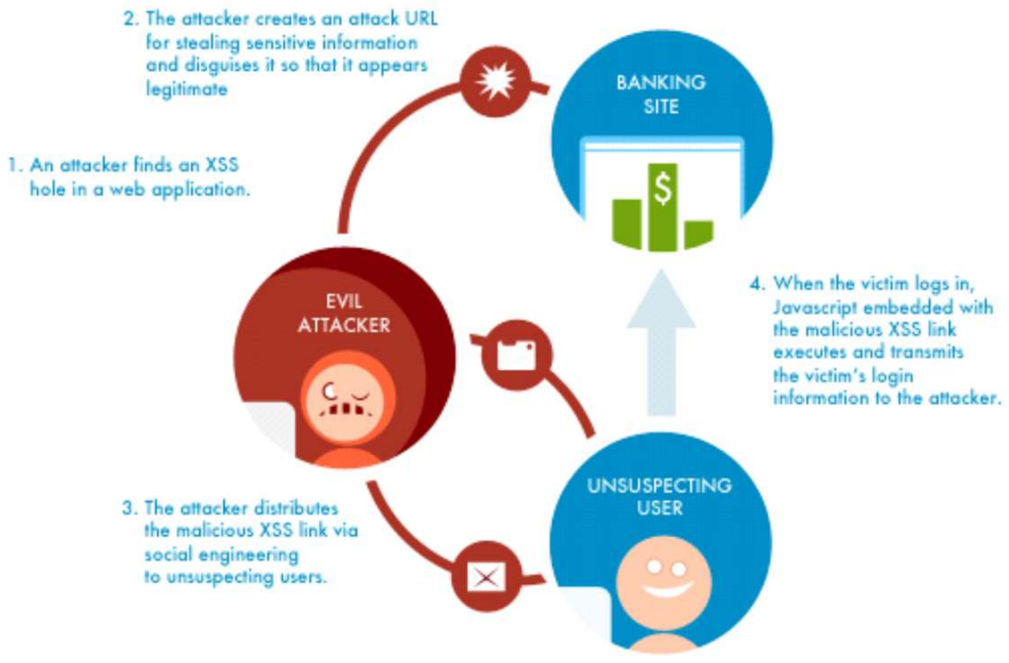
Server: http://10.100.92.99:80



Critical

Cross-Site Scripting: Reflected

Summary:


2. The attacker creates an attack URL for stealing sensitive information and disguises it so that it appears legitimate
BANKING SITE
1. An attacker finds an XSS hole in a web application.
EVIL ATTACKER
4. When the victim logs in, Javascript embedded with the malicious XSS link executes and transmits the victim's login information to the attacker.
UNSUSPECTING USER
3. The attacker distributes the malicious XSS link via social engineering to unsuspecting users.

Cross-Site Scripting vulnerability found in Get parameter status. The following attack uses plain encoding:

<sCrIpT>alert(21853)</sCrIpT>

Cross-Site Scripting vulnerabilities were verified as executing code on the web application. Cross-Site Scripting occurs when dynamically generated web pages display user input, such as login information, that is not properly validated, allowing an attacker to embed malicious scripts into the generated page and then execute the script on the machine of any user that views the site. In this instance, the web application was vulnerable to an automatic payload, meaning the user simply has to visit a page to make the malicious scripts execute. If successful, Cross-Site Scripting vulnerabilities can be exploited to manipulate or steal cookies, create requests that can be mistaken for those of a valid user, compromise confidential information, or execute malicious code on end user systems. Recommendations include implementing secure programming techniques that ensure proper filtration of user-supplied data, and encoding all user supplied data to prevent inserted scripts being sent to end users in a format that can be executed.

Execution:

View the attack string included with the request to check what to search for in the response. For instance, if "(javascript:alert ('XSS')" is submitted as an attack (or another scripting language), it will also appear as part of the response. This indicates that the web application is taking values from the HTTP request parameters and using them in the HTTP response without first removing potentially malicious data.

Implication:

XSS can generally be subdivided into two categories: stored and reflected attacks. The main difference between the two is in how the payload arrives at the server. Stored attacks are just that...in some form stored on the target server, such as in a database, or via a submission to a bulletin board or visitor log. The victim will retrieve and execute the attack code in his browser when a request is made for the stored information. Reflected attacks, on the other hand, come from somewhere else. This happens when user input from a web client is immediately included via server-side scripts in a dynamically generated web page. Via some social engineering, an attacker can trick a victim, such as through a malicious link or "rigged" form, to submit information which will be altered to include attack code and then sent to the legitimate server. The injected code is then reflected back to the user's browser which executes it because it came from a trusted server. The implication of each kind of attack is the same.

The main problems associated with successful Cross-Site Scripting attacks are:

- Account hijacking - An attacker can hijack the user's session before the session cookie expires and take actions with the privileges of the user who accessed the URL, such as issuing database queries and viewing the results.
- Malicious script execution - Users can unknowingly execute JavaScript, VBScript, ActiveX, HTML, or even Flash content that has been inserted into a dynamically generated page by an attacker.
- Worm propagation - With Ajax applications, XSS can propagate somewhat like a virus. The XSS payload can autonomously inject itself into pages, and easily re-inject the same host with more XSS, all of which can be done with no hard refresh. Thus, XSS can send multiple requests using complex HTTP methods to propagate itself invisibly to the user.
- Information theft - Via redirection and fake sites, attackers can connect users to a malicious server of the attacker's choice and capture any information entered by the user.
- Denial of Service - Often by utilizing malformed display requests on sites that contain a Cross-Site Scripting vulnerability, attackers can cause a denial of service condition to occur by causing the host site to query itself repeatedly .
- Browser Redirection - On certain types of sites that use frames, a user can be made to think that he is in fact on the original site when he has been redirected to a malicious one, since the URL in the browser's address bar will remains the same. This is because the entire page isn't being redirected, just the frame in which the JavaScript is being executed.
- Manipulation of user settings - Attackers can change user settings for nefarious purposes.

For more detailed information on Cross-Site Scripting attacks, see the HP Cross-Site Scripting whitepaper.

Fix:

For Development:

Cross-Site Scripting attacks can be avoided by carefully validating all input, and properly encoding all output. When validating user input, verify that it matches the strictest definition of valid input possible. For example, if a certain parameter is supposed to be a number, attempt to convert it to a numeric data type in your programming language.

PHP: intval("0".$_GET['q']);

ASP.NET: int.TryParse(Request.QueryString["q"], out val);

The same applies to date and time values, or anything that can be converted to a stricter type before being used. When accepting other types of text input, make sure the value matches either a list of acceptable values (white-listing), or a strict regular expression. If at any point the value appears invalid, do not accept it. Also, do not attempt to return the value to the user in an error message.

Most server side scripting languages provide built in methods to convert the value of the input variable into correct, non-interpretable HTML. These should be used to sanitize all input before it is displayed to the client.

PHP: string htmlspecialchars (string string [, int quote_style])

ASP.NET: Server.HTMLEncode (strHTML String)

When reflecting values into JavaScript or another format, make sure to use a type of encoding that is appropriate. Encoding data for HTML is not sufficient when it is reflected inside of a script or style sheet. For example, when reflecting data in a JavaScript string, make sure to encode all non-alphanumeric characters using hex (\xHH) encoding.

If you have JavaScript on your page that accesses unsafe information (like location.href) and writes it to the page (either with document.write, or by modifying a DOM element), make sure you encode data for HTML before writing it to the page. JavaScript does not have a built-in function to do this, but many frameworks do. If you are lacking an available function, something like the following will handle most cases:

```
s = s.replace(/&/g,'&amp;').replace(/"/i,'&quot;').replace(/</i,'&lt;').replace(/>/i,'&gt;').replace(/'/i,'&apos;')
```

Ensure that you are always using the right approach at the right time. Validating user input should be done as soon as it is received. Encoding data for display should be done immediately before displaying it.

For Security Operations:

Server-side encoding, where all dynamic content is first sent through an encoding function where Scripting tags will be replaced with codes in the selected character set, can help to prevent Cross-Site Scripting attacks.

Many web application platforms and frameworks have some built-in support for preventing Cross-Site Scripting. Make sure that any built-in protection is enabled for your platform. In some cases, a misconfiguration could allow Cross-Site Scripting. In ASP.NET, if a page's EnableViewStateMac property is set to False, the ASP.NET view state can be used as a vector for Cross-Site Scripting.

An IDS or IPS can also be used to detect or filter out XSS attacks. Below are a few regular expressions that will help detect Cross-Site Scripting.

Regex for a simple XSS attack:

```
/((\%3C) <)((\%2F) \/)*[a-z0-9\%]+((\%3E) >)/ix
```

The above regular expression would be added into a new Snort rule as follows:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"NII Cross-Site Scripting attempt";
```

flow:to_server,established; pcre:"/((\%3C) <)((\%2F) \/)*[a-z0-9\%]+((\%3E) >)/i"; classtype:Web-application-attack; sid:9000; rev:5;)

Paranoid regex for XSS attacks:
/((\%3C) <)[^\n]+((\%3E) >)/I

This signature simply looks for the opening HTML tag, and its hex equivalent, followed by one or more characters other than the new line, and then followed by the closing tag or its hex equivalent. This may end up giving a few false positives depending upon how your web application and web server are structured, but it is guaranteed to catch anything that even remotely resembles a Cross-Site Scripting attack.

For QA:

Fixes for Cross-Site Scripting defects will ultimately require code based fixes. Read the HP Cross-Site Scripting white paper for more information about manually testing your application for Cross-Site Scripting.

Reference:

HP Cross-Site Scripting Whitepaper
http://download.hpsmartupdate.com/asclabs/cross-site_scripting.pdf

OWASP Cross-Site Scripting Information
http://www.owasp.org/documentation/topten/a4.html

Microsoft
http://support.microsoft.com/default.aspx?scid=kb;EN-US;q252985

Microsoft Anti-Cross Site Scripting Library V1.0
http://www.microsoft.com/downloads/details.aspx?familyid=9a2b9c92-7ad9-496c-9a89-af08de2e5982&displaylang=en

CERT
http://www.cert.org/advisories/CA-2000-02.html

Apache
http://httpd.apache.org/info/css-security/apache_specific.html

SecurityFocus.com
http://www.securityfocus.com/infocus/1768

File Names:

- http://10.100.92.99:80/frontend/web/student/activity/training-student-status.aspx?status=4d673d3d%3c
- http://10.100.92.99:80/frontend/web/student/activity/property.aspx?training_id=4f44493d"><sCrIpT>win
- http://10.100.92.99:80/frontend/web/student/default/index.aspx?ActivitySearch%5Bname%5D=12345&Activi
- http://10.100.92.99:80/frontend/web/student/activity/class-student.aspx?training_id=82&class_id=38&p
- http://10.100.92.99:80/frontend/web/student/default/index.aspx?ActivitySearch%5Bname%5D=12345&Activi
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?ActivitySearch%5Bname%5D=12345&Activ
- http://10.100.92.99:80/frontend/web/student/activity/class.aspx?training_id=4f44493d&training_studen
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?ActivitySearch%5Bname%5D=12345&Activ
- http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx?training_id=4
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?satker_id=&year=2015&_pjax=%23pjax-g
- http://10.100.92.99:80/frontend/web/student/activity/student.aspx?training_id=4f44493d&training_stud
- http://10.100.92.99:80/frontend/web/student/activity/property.aspx?training_id=4f44493d"><sCrIpT>win
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?sort=-end--></sCrIpT><sCrIpT>alert(6
- http://10.100.92.99:80/frontend/web/student/activity/property.aspx?training_id=4f44493d&training_stu
- http://10.100.92.99:80/frontend/web/student/activity/training-student-status.aspx?

status=4d673d3d&tr

High **Credential Management: Insecure Transmission**

Summary:

Any area of a web application that possibly contains sensitive information or access to privileged functionality such as remote site administration functionality should utilize SSL or another form of encryption to prevent login information from being sniffed or otherwise intercepted or stolen. http://10.100.92.99:80/frontend/web/site/masuk.aspx has failed this policy. Recommendations include ensuring that sensitive areas of your web application have proper encryption protocols in place to prevent login information and other data that could be helpful to an attacker from being intercepted.

Implication:

An attacker who exploited this design vulnerability would be able to utilize the information to escalate their method of attack, possibly leading to impersonation of a legitimate user, the theft of proprietary data, or execution of actions not intended by the application developers.

Fix:

For Security Operations:
Ensure that sensitive areas of your web application have proper encryption protocols in place to prevent login information and other data that could be helpful to an attacker from being intercepted.

For Development:
Ensure that sensitive areas of your web application have proper encryption protocols in place to prevent login information and other data that could be helpful to an attacker from being intercepted.

For QA:
Test the application not only from the perspective of a normal user, but also from the perspective of a malicious one.

File Names:

- http://10.100.92.99:80/frontend/web/site/masuk.aspx

High **Transport Layer Protection: Unencrypted Login Form**

Summary:

An unencrypted login form has been discovered. Any area of a web application that possibly contains sensitive information or access to privileged functionality such as remote site administration functionality should utilize SSL or another form of encryption to prevent login information from being sniffed or otherwise intercepted or stolen. If the login form is being served over SSL, the page that the form is being submitted to MUST be accessed over SSL. Every link/URL present on that page (not just the form action) needs to be served over HTTPS. This will prevent Man-in-the-Middle attacks on the login form. Recommendations include ensuring that sensitive areas of your web application have proper encryption protocols in place to prevent login information and other data that could be helpful to an attacker from being intercepted.

Implication:

An attacker who exploited this design vulnerability would be able to utilize the information to escalate their method of attack, possibly leading to impersonation of a legitimate user, the theft of proprietary data, or execution of actions not intended by the application developers.

Fix:

Ensure that sensitive areas of your web application have proper encryption protocols in place to prevent login information and other data that could be helpful to an attacker from being intercepted.

Reference:

Advisory:http://www.kb.cert.org/vuls/id/466433

File Names:

- http://10.100.92.99:80/frontend/web/site/masuk.aspx

High **Cross-Frame Scripting**

Summary:

A Cross-Frame Scripting (XFS) vulnerability can allow an attacker to load the vulnerable application inside an HTML iframe tag on a malicious page. The attacker could use this weakness to devise a Clickjacking attack to conduct phishing, frame sniffing, social engineering or Cross-Site Request Forgery attacks.

Clickjacking
The goal of a Clickjacking attack is to deceive the victim user into interacting with UI elements of the attacker's choice on the

target web site without her knowledge and in turn executing privileged functionality on the victim’s behalf. To achieve this goal, the attacker must exploit the XFS vulnerability to load the attack target inside an iframe tag, hide it using Cascading Style Sheets (CSS) and overlay the phishing content on the malicious page. By placing the UI elements on the phishing page to overlap with those on the page targeted in the attack, the attacker can ensure that the victim is forced to interact with the UI elements on the target page not visible to the victim.
WebInspect has detected a page which potentially handles sensitive information using an HTML form with a password input field and is missing XFS protection.

This response is not protected by a valid X-Frame-Options header.Furthermore,
An effective frame-busting technique was not observed while loading this page inside a frame.

Execution:

Create a test page containing an HTML iframe tag whose src attribute is set to http://10.100.92.99:80/frontend/web/site/masuk.aspx. Successful framing of the target page indicates the application’s susceptibility to XFS.
Note that WebInspect will report only one instance of this check across each host within the scope of the scan. The other visible pages on the site may, however, be vulnerable to XFS as well and hence should be protected against it with an appropriate fix.

Implication:

A Cross-Frame Scripting weakness could allow an attacker to embed the vulnerable application inside an iframe. Exploitation of this weakness could result in:

Hijacking of user events such as keystrokes
Theft of sensitive information
Execution of privileged functionality through combination with Cross-Site Request Forgery attacks

Fix:

Browser vendors have introduced and adopted a policy-based mitigation technique using the X-Frame-Options header. Developers can use this header to instruct the browser about appropriate actions to perform if their site is included inside an iframe.Developers must set the X-Frame-Options header to one of the following permitted values:

- DENY

Deny all attempts to frame the page

- SAMEORIGIN

The page can be framed by another page only if it belongs to the same origin as the page being framed

- ALLOW-FROM origin

Developers can specify a list of trusted origins in the origin attribute. Only pages on origin are permitted to load this page inside an iframe

Developers must **also** use client-side frame busting JavaScript as a protection against XFS. This will enable users of older browsers that do not support the X-Frame-Options header to also be protected from clickjacking attacks.

Reference:

HP 2012 Cyber Security Report
The X-Frame-Options header - a failure to launch

Server Configuration:
IIS
Apache, nginx

Specification:
X-Frame-Options IETF Draft

OWASP:
Clickjacking

Frame Busting:
Busting Frame Busting: A Study of Clickjacking Vulnerabilities on Popular Sites
OWASP: Busting Frame Busting

File Names:
- http://10.100.92.99:80/frontend/web/site/masuk.aspx

Medium **Directory Listing**

Summary:

A serious Directory Listing vulnerability was discovered within your web application. Risks associated with an attacker discovering a Directory Listing, which is a complete index of all of the resources located in that directory, result from the fact that files that should remain hidden, such as data files, backed-up source code, or applications in development, may then be visible. The specific risks depend upon the specific files that are listed and accessible. Recommendations include restricting access to important directories or files by adopting a "need to know" requirement for both the document and server root, and turning off features such as Automatic Directory Listings that could expose private files and provide information that could be utilized by an attacker when formulating or conducting an attack.

Execution:

http://10.100.92.99:80/frontend/web/assets/189fe31c/

Implication:

Risks associated with an attacker discovering a Directory Listing on your application server depend upon what type of directory is discovered, and what types of files are contained within it. The primary threat from an accessible Directory Listing is that hidden files such as data files, source code, or applications under development will then be visible to a potential attacker. In addition to accessing files containing sensitive information, other risks include an attacker utilizing the information discovered in that directory to perform other types of attacks.

Fix:

For Development:

you are actively involved with implementing the web application server, there is not a wide range of available solutions to prevent problems that can occur from an attacker finding a Directory Listing. Primarily, this problem will be resolved by the web application server administrator. However, there are certain actions you can take that will help to secure your web application.

- Restrict access to important files or directories only to those who actually need it.
- Ensure that files containing sensitive information are not left publicly accessible, or that comments left inside files do not reveal the locations of directories best left confidential.

For Security Operations:

One of the most important aspects of web application security is to restrict access to important files or directories only to those individuals who actually need to access them. Ensure that the private architectural structure of your web application is not exposed to anyone who wishes to view it as even seemingly innocuous directories can provide important information to a potential attacker.

The following recommendations can help to ensure that you are not unintentionally allowing access to either information that could be utilized in conducting an attack or propriety data stored in publicly accessible directories.

- Turn off the Automatic Directory Listing feature in whatever application server package that you utilize.
- Restrict access to important files or directories only to those who actually need it.
- Ensure that files containing sensitive information are not left publicly accessible.
- Don't follow standard naming procedures for hidden directories. For example, don't create a hidden directory called "cgi" that contains cgi scripts. Obvious directory names are just that...readily guessed by an attacker.

Remember, the harder you make it for an attacker to access information about your web application, the more likely it is that he will simply find an easier target.

For QA:

For reasons of security, it is important to test the web application not only from the perspective of a normal user, but also from that of a malicious one. Whenever possible, adopt the mindset of an attacker when testing your web application for security defects. Access your web application from outside your firewall or IDS. Utilize Google or another search engine to ensure that searches for vulnerable files do not return information from regarding your web application. For example, an attacker will utilize a search engine, and search for directory listings such as the following: "index of / cgi-bin". Make sure that your directory structure is not obvious, and that only files that are necessary are capable of being accessed.

Reference:

Apache:
Security Tips for Server Configuration
Protecting Confidential Documents at Your Site
Securing Apache - Access Control

IIS:
Implementing NTFS Standard Permissions on Your Web Site

Netscape:
Controlling Access to Your Server

General:
Password-protecting web pages
Web Security

File Names:

- http://10.100.92.99:80/frontend/web/assets/189fe31c/
- http://10.100.92.99:80/frontend/web/assets/47807c76/css/
- http://10.100.92.99:80/frontend/models/
- http://10.100.92.99:80/frontend/web/assets/ea5317b3/js/
- http://10.100.92.99:80/frontend/web/assets/f00db76d/img/
- http://10.100.92.99:80/frontend/runtime/logs/
- http://10.100.92.99:80/frontend/web/assets/73de47c/css/
- http://10.100.92.99:80/frontend/web/assets/73de47c/img/
- http://10.100.92.99:80/frontend/web/assets/4b8dd310/
- http://10.100.92.99:80/frontend/views/layouts/
- http://10.100.92.99:80/frontend/web/assets/73de47c/js/
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/unit/
- http://10.100.92.99:80/frontend/web/assets/6538843c/
- http://10.100.92.99:80/frontend/web/assets/d021eecb/
- http://10.100.92.99:80/frontend/runtime/
- http://10.100.92.99:80/frontend/web/assets/8969b36/
- http://10.100.92.99:80/frontend/web/assets/ace9a976/fonts/
- http://10.100.92.99:80/frontend/web/assets/ace9a976/
- http://10.100.92.99:80/frontend/web/assets/ace9a976/js/
- http://10.100.92.99:80/frontend/messages/
- http://10.100.92.99:80/frontend/web/assets/b44ebcff/
- http://10.100.92.99:80/frontend/web/assets/9451fef9/
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/
- http://10.100.92.99:80/frontend/web/assets/b44ebcff/img/
- http://10.100.92.99:80/frontend/modules/student/models/
- http://10.100.92.99:80/frontend/web/assets/4b8dd310/css/
- http://10.100.92.99:80/frontend/runtime/cache/
- http://10.100.92.99:80/frontend/web/assets/73de47c/
- http://10.100.92.99:80/frontend/modules/student/controllers/
- http://10.100.92.99:80/frontend/web/assets/4b8dd310/js/
- http://10.100.92.99:80/frontend/web/assets/47807c76/js/
- http://10.100.92.99:80/frontend/web/assets/b604c748/css/
- http://10.100.92.99:80/frontend/modules/
- http://10.100.92.99:80/frontend/web/assets/8969b36/script/
- http://10.100.92.99:80/frontend/web/assets/9451fef9/js/
- http://10.100.92.99:80/frontend/web/assets/8969b36/swf/
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/
- http://10.100.92.99:80/frontend/modules/student/views/student/

- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/
- http://10.100.92.99:80/frontend/web/assets/16926e51/
- http://10.100.92.99:80/frontend/template_ereg/
- http://10.100.92.99:80/frontend/web/assets/ea5317b3/
- http://10.100.92.99:80/frontend/
- http://10.100.92.99:80/frontend/web/assets/16926e51/js/locales/
- http://10.100.92.99:80/frontend/runtime/debug/
- http://10.100.92.99:80/frontend/web/assets/8969b36/audio/
- http://10.100.92.99:80/frontend/web/assets/9b922f9d/
- http://10.100.92.99:80/frontend/web/assets/e2cb43ee/css/
- http://10.100.92.99:80/frontend/web/assets/47807c76/
- http://10.100.92.99:80/frontend/web/assets/f00db76d/css/
- http://10.100.92.99:80/frontend/web/assets/47807c76/img/
- http://10.100.92.99:80/frontend/views/
- http://10.100.92.99:80/frontend/web/assets/f00db76d/js/
- http://10.100.92.99:80/frontend/web/assets/e2cb43ee/js/
- http://10.100.92.99:80/frontend/web/assets/e2cb43ee/img/
- http://10.100.92.99:80/frontend/messages/id-ID/
- http://10.100.92.99:80/frontend/runtime/cache/96/
- http://10.100.92.99:80/frontend/web/assets/ace9a976/css/
- http://10.100.92.99:80/frontend/web/assets/f00db76d/
- http://10.100.92.99:80/frontend/controllers/
- http://10.100.92.99:80/frontend/web/assets/16926e51/css/
- http://10.100.92.99:80/frontend/widgets/
- http://10.100.92.99:80/frontend/web/assets/16926e51/js/
- http://10.100.92.99:80/frontend/modules/student/
- http://10.100.92.99:80/frontend/modules/student/views/
- http://10.100.92.99:80/frontend/web/assets/e2cb43ee/
- http://10.100.92.99:80/frontend/web/assets/b604c748/fonts/
- http://10.100.92.99:80/frontend/config/
- http://10.100.92.99:80/frontend/web/assets/b44ebcff/css/
- http://10.100.92.99:80/frontend/web/assets/b44ebcff/js/
- http://10.100.92.99:80/frontend/web/assets/b604c748/
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/

Medium **Cookie Security: Persistent Cookie**

Summary:

Cookies are small bits of data that are sent by the web application but stored locally in the browser. This lets the application use the cookie to pass information between pages and store variable information. The web application controls what information is stored in a cookie and how it is used. Typical types of information stored in cookies are session Identifiers, personalization and customization information, and in rare cases even usernames to enable automated logins. There are two different types of cookies: *session cookies*and *persistent cookies*. Session cookies only live in the browser's memory, and are not stored anywhere. Persistent cookies, however, are stored on the browser's hard drive. This can cause security and privacy issues depending on the information stored in the cookie and how it is accessed.

Execution:

All cookies are set by the server via the Set-Cookie HTTP Header. A browser knows to store that cookie as a persistent cookie when it finds the keyword 'Expires=' followed by a date in the future. If there is no 'Expires=' tag, or if the specified date has already passed, then the browser will keep the cookie in memory only as a session cookie.

To view the persistent cookie set on this page, view the **HTTP response** and examine the Set-Cookie header. You should see the 'Expires=' tag with a future date specified.

Implication:

Persistent cookies are stored on the browsing clients hard drive even when that client is no longer browsing the Web site that set the client. Depending on what information is stored in the cookie, this could lead to security and privacy violations. The Office of Management and Budget has decreed that no federal websites shall use persistent cookies except in very specific situations.

Fix:

From a coding perspective, the only distinction between a session cookie and a persistent cookie is the 'Expires=' tag that specifies when a persistent cookie should expire. If a cookie has no 'Expires=' tag, then it is automatically interpreted as a session cookie. Removing the expiration date from the code that sets the cookie will change it to a session cookie.

Reference:

White House Office of Management and Budget:
Memorandum M-00-13Privacy Policies and Data Collection on Federal Web Sites

Microsoft Knowledgebase Article:
Description of Persistent and Per-Session Cookies in Internet Explorer.

File Names:
- http://10.100.92.99:80/frontend/web/site/masuk.aspx

Medium **Privacy Violation: Autocomplete**

Summary:

Most recent browsers have features that will save password field content entered by users and then automatically complete password entry the next time the field are encountered. This feature is enabled by default and could leak password since it is stored on the hard drive of the user. The risk of this issue is greatly increased if users are accessing the application from a shared environment. Recommendations include setting autocomplete to "off" on all your password fields.

Execution:

To verify if a password filed is vulnerable, first make sure to enable the autocomplete in your browser's settings, and then input the other fileds of the form to see whether the password is automatically filled. If yes, then it's vulnerable, otherwise, not. You may need to do it twice in case it is the first time you type in the credential in your browser.

Implication:

When autocomplete is enabled, hackers can directly steal your password from local storage.

Fix:

From the web application perspective, the autocomplete can be turned at the form level or individual entry level by defining the attribute AUTOCOMPLETE="off".

Reference:

Microsoft:
Autocomplete Security

File Names:
- http://10.100.92.99:80/frontend/web/student/student/password.aspx

---

Medium **Cross-Frame Scripting**

Summary:

A Cross-Frame Scripting (XFS) vulnerability can allow an attacker to load the vulnerable application inside an HTML iframe tag on a malicious page. The attacker could use this weakness to devise a Clickjacking attack to conduct phishing, frame sniffing, social engineering or Cross-Site Request Forgery attacks.

Clickjacking
The goal of a Clickjacking attack is to deceive the victim user into interacting with UI elements of the attacker’s choice on the target web site without her knowledge and in turn executing privileged functionality on the victim’s behalf. To achieve this goal, the attacker must exploit the XFS vulnerability to load the attack target inside an iframe tag, hide it using Cascading Style Sheets (CSS) and overlay the phishing content on the malicious page. By placing the UI elements on the phishing page to overlap with those on the page targeted in the attack, the attacker can ensure that the victim is forced to interact with the UI elements on the target page not visible to the victim.
WebInspect has detected a response containing one or more forms that accept user input but is missing XFS protection.

This response is not protected by a valid X-Frame-Options header.Furthermore,
An effective frame-busting technique was not observed while loading this page inside a frame.

Execution:

Create a test page containing an HTML <iframe> tag whose **src** attribute is set to http://10.100.92.99:80/frontend/web/student/student/profile.aspx. Successful framing of the target page indicates the

application's susceptibility to XFS.
Note that WebInspect will report only one instance of this check across each host within the scope of the scan. The other visible pages on the site may, however, be vulnerable to XFS as well and hence should be protected against it with an appropriate fix.

Implication:

A Cross-Frame Scripting weakness could allow an attacker to embed the vulnerable application inside an iframe. Exploitation of this weakness could result in:

Hijacking of user events such as keystrokes
Theft of sensitive information
Execution of privileged functionality through combination with Cross-Site Request Forgery attacks

Fix:

Browser vendors have introduced and adopted a policy-based mitigation technique using the X-Frame-Options header. Developers can use this header to instruct the browser about appropriate actions to perform if their site is included inside an iframe. Developers must set the X-Frame-Options header to one of the following permitted values:

- DENY

Deny all attempts to frame the page

- SAMEORIGIN

The page can be framed by another page only if it belongs to the same origin as the page being framed

- ALLOW-FROM origin

Developers can specify a list of trusted origins in the origin attribute. Only pages on origin are permitted to load this page inside an iframe

Developers must **also** use client-side frame busting JavaScript as a protection against XFS. This will enable users of older browsers that do not support the X-Frame-Options header to also be protected from clickjacking attacks.

Reference:

HP 2012 Cyber Security Report
The X-Frame-Options header - a failure to launch

Server Configuration:
IIS
Apache, nginx

Specification:
X-Frame-Options IETF Draft

OWASP:
Clickjacking

Frame Busting:
Busting Frame Busting: A Study of Clickjacking Vulnerabilities on Popular Sites
OWASP: Busting Frame Busting

File Names:

- http://10.100.92.99:80/frontend/web/student/student/profile.aspx

Low

Poor Error Handling: Unhandled Exception

Summary:

A minor vulnerability has been discovered within your web application due to the the presence of a fully qualified path name to the root of your system. This most often occurs in context of an error being produced by the web application. Fully qualified server path names allow an attacker to know the file system structure of the web server, which is a baseline for many other types of attacks to be successful. Recommendations include adopting a consistent error handling scheme and mechanism that prevents fully qualified path names from being displayed.

Execution:

To verify the issue, click the 'HTTP Response' button on the properties view and review the highlighted areas to determine the Unix path found.

Fix:

For Development:

Don't display fully qualified pathnames as part of error or informational messages. At the least, fully qualified pathnames can provide an attacker with important information about the architecture of web application.

For Security Operations:

The following recommendations will help to ensure that a potential attacker is not deriving valuable information from any error message that is presented.

- **Uniform Error Codes:** Ensure that you are not inadvertently supplying information to an attacker via the use of inconsistent or "conflicting" error messages. For instance, don't reveal unintended information by utilizing error messages such as Access Denied, which will also let an attacker know that the file he seeks actually exists. Have consistent terminology for files and folders that do exist, do not exist, and which have read access denied.

- **Informational Error Messages:** Ensure that error messages do not reveal too much information. Complete or partial paths, variable and file names, row and column names in tables, and specific database errors should never be revealed to the end user. Remember, an attacker will gather as much information as possible, and then add pieces of seemingly innocuous information together to craft a method of attack.

- **Proper Error Handling:** Utilize generic error pages and error handling logic to inform end users of potential problems. Do not provide system information or other data that could be utilized by an attacker when orchestrating an attack.

For QA:

In reality, simple testing can usually determine how your web application will react to different input errors. More expansive testing must be conducted to cause internal errors to gauge the reaction of the site.

The best course of action for QA associates to take is to ensure that the error handling scheme is consistent. Do you receive a different type of error for a file that does not exist as opposed to a file that does? Are phrases like "Permission Denied" utilized which could reveal the existence of a file to an attacker? It is often a seemingly innocuous piece of information that provides an attacker with the means to discover something else which he can then utilize when conducting an attack.

File Names:

- http://10.100.92.99:80/frontend/runtime/debug/54e066fec56e2.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f6f32f33.data
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=4f44493
- http://10.100.92.99:80/frontend/runtime/logs/app.log
- http://10.100.92.99:80/frontend/runtime/debug/54e06b846c523.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b847a78b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068ca8b2ac.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d361d20.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f1a8046d.data
- http://10.100.92.99:80/frontend/web/student/student/profile.aspx
- http://10.100.92.99:80/frontend/runtime/debug/54e066feeb3f3.data
- http://10.100.92.99:80/frontend/runtime/debug/54e186c3455c0.data
- http://10.100.92.99:80/frontend/runtime/debug/54e069434976f.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f08daa5f.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ff4af4c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068cad2529.data
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx
- http://10.100.92.99:80/frontend/runtime/debug/54e068d40a509.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f0ddaee6.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155cc17a09.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b69d45ac.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b699d6f8.data
- http://10.100.92.99:80/frontend/web/student/activity/training-student-status.aspx

- http://10.100.92.99:80/frontend/runtime/debug/54e063c38dcc1.data
- http://10.100.92.99:80/frontend/runtime/debug/54e0637656735.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066e73251c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b6951238.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06cb778a42.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d3b519c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155851deaf.data
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx
- http://10.100.92.99:80/frontend/runtime/debug/54e066ffa08d0.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d3cdf9e.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f0e1995b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f1a428d9.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b6c1e01c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ca75c8f7.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06cab7eb87.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce1a82ee.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce15ac44.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066e6c249f.data
- http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx
- http://10.100.92.99:80/frontend/runtime/debug/54e069438de91.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06954e6e7d.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f0833ad2.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f08778a5.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155c810dc9.data
- http://10.100.92.99:80/frontend/web/assets/8969b36/script/soundmanager2.js
- http://10.100.92.99:80/frontend/runtime/debug/54e06b83e1d23.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b8469e0b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b841634b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ff4b217.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ff6fc66.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b69768b7.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce1d77f1.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155c7da00c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e186c0d216f.data
- http://10.100.92.99:80/frontend/web/student/student/profile.aspx
- http://10.100.92.99:80/frontend/web/student/student/profile.aspx
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce1803ce.data
- http://10.100.92.99:80/frontend/runtime/logs/app.log.1
- http://10.100.92.99:80/frontend/runtime/debug/54e06343a7091.data
- http://10.100.92.99:80/frontend/web/assets/9b922f9d/release.sh
- http://10.100.92.99:80/frontend/runtime/debug/54e068d3b555d.data
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=4f44493d&
- http://10.100.92.99:80/frontend/runtime/debug/54e06b84b4c98.data
- http://10.100.92.99:80/frontend/web/assets/d021eecb/release.sh

- http://10.100.92.99:80/frontend/runtime/debug/54e066ea9220d.data

Low **Often Misused: File Upload**

Summary:

An indicator of file upload capability was found. File upload capability allows a web user to send a file from his or her computer to the webserver. If the web application that receives the file does not carefully examine it for malicious content, an attacker may be able to use file uploads to execute arbitrary commands on the server. Recommendations include adopting a strict file upload policy that prevents malicious material from being uploaded via sanitization and filtering.

Implication:

The exact implications depend upon the nature of the files an attacker would be able to upload. Implications range from unauthorized content publishing to aid in phising attacks, all the way to full compromise of the web server.

Fix:

For Security Operations:
This check is part of unknown application testing. Unknown application testing seeks to uncover new vulnerabilities in both custom and commercial software. Because of this, there are no specific patches or descriptions for this issue. If there is no apparent file upload capability on the page, this check may be safely ignored. You can instruct the scanner to ignore this vulnerability by right-clicking the vulnerability node on the displayed results tree and click "Ignore Vulnerability."

For QA:
This issue will need to be resolved in the production code. Notify the appropriate developer of this issue.

For Development:
Ensure that the following steps are taken to sanitize the file being received:

- Limit the types of files that can be uploaded. For instance, on an image upload page, any file other than a .jpg should be refused.
- Ensure that the web user has no control whatsoever over the name and location of the uploaded file on the server.
- Never use the name that the user assigns it.
- Never derive the filename from the web user's username or session ID.
- Do not place the file in a directory accessible by web users. It is preferable for this location to be outside of the webroot.
- Ensure that strict permissions are set on both the uploaded file and the directory it is located in.
- Do not allow execute permissions on uploaded files. If possible, deny all permission for all users but the web application user.
- Verify that the uploaded file contains appropriate content. For instance, an uploaded JPEG should have a standard JPEG file header.

File Names:

- http://10.100.92.99:80/frontend/web/assets/b44ebcff/README.md
- http://10.100.92.99:80/frontend/web/student/student/profile.aspx
- http://10.100.92.99:80/frontend/web/assets/b44ebcff/examples/

Low **Access Control: Unprotected File**

Summary:

System Environment variables log files contain information about the nature of your web application, and would allow an attacker to gain insightful information about the web system setup. Recommendations include removing this file from the affected system.

Implication:

A fundamental part of any successful attack is reconnaissance and information gathering. The primary danger from exploitation of this vulnerability is that an attacker will be able to utilize the information in launching a more serious attack. It is very simple to check for its existence, and a file most definitely on the short list of things for which a potential attacker would look.

Fix:

For Security Operations:
Remove this file from the system in question. One of the most important aspects of web application security is to restrict access to important files or directories only to those individuals who actually need to access them. Ensure that the private architectural structure of your web application is not exposed to anyone who wishes to view it as even seemingly innocuous

directories can provide important information to a potential attacker.

For QA:
Notify your Security or Network Operations team of this issue.

For Development:
Notify your Security or Network Operations team of this issue.

File Names:

- http://10.100.92.99:80/frontend/runtime/debug/54e06b6c1e01c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b699d6f8.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d40a509.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce1d77f1.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f0833ad2.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f1a8046d.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155c7da00c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e186c0d216f.data
- http://10.100.92.99:80/frontend/web/file/download.aspx?file=%00
- http://10.100.92.99:80/frontend/web/student/activity/training-student-status.aspx?status=4d673d3d&tr
- http://10.100.92.99:80/frontend/web/student/student/profile.aspx
- http://10.100.92.99:80/frontend/runtime/debug/54e068d3b555d.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ff4b217.data
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=4f44493d&
- http://10.100.92.99:80/frontend/runtime/debug/54e14f0e1995b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06954e6e7d.data
- http://10.100.92.99:80/frontend/web/student/activity/training-student-status.aspx?status=4d673d3d%26
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=http://ww
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=http://
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=4f44493d&
- http://10.100.92.99:80/frontend/runtime/debug/54e066e73251c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06cb778a42.data
- http://10.100.92.99:80/frontend/runtime/debug/54e063c38dcc1.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b846c523.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068ca8b2ac.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b6951238.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce1803ce.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06343a7091.data
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx
- http://10.100.92.99:80/frontend/runtime/debug/54e155cc17a09.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066feeb3f3.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ff4af4c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f08daa5f.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f08778a5.data
- http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx
- http://10.100.92.99:80/frontend/runtime/logs/app.log
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=4f44493

- http://10.100.92.99:80/frontend/web/student/activity/training-student-status.aspx
- http://10.100.92.99:80/frontend/runtime/debug/54e06b69d45ac.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b84b4c98.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ea9220d.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155851deaf.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce15ac44.data
- http://10.100.92.99:80/frontend/runtime/debug/54e069438de91.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f1a428d9.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ff6fc66.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ffa08d0.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b8469e0b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b69768b7.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b841634b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ca75c8f7.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06cab7eb87.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce1a82ee.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155c810dc9.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d3cdf9e.data
- http://10.100.92.99:80/frontend/runtime/debug/54e069434976f.data
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=4f44493
- http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx?training_id=h
- http://10.100.92.99:80/frontend/runtime/debug/54e068d3b519c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b847a78b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e186c3455c0.data
- http://10.100.92.99:80/frontend/runtime/debug/54e0637656735.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066e6c249f.data
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=4f44493d&
- http://10.100.92.99:80/frontend/runtime/debug/54e066fec56e2.data
- http://10.100.92.99:80/frontend/runtime/logs/app.log.1
- http://10.100.92.99:80/frontend/runtime/debug/54e06b83e1d23.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f6f32f33.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d361d20.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f0ddaee6.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068cad2529.data
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=4f44493

Low

System Information Leak: Internal IP

Summary:

A string matching an internal/reserved IPv4 or IPv6 address range was discovered. This may disclose information about the IP addressing scheme of the internal network and can be valuable to attackers.Internal IPv4/IPv6 ranges are:
10.x.x.x
172.16.x.x through 172.31.x.x
192.168.x.x
fd00::x
If not a part of techical documentation, recommendations include removing the string from the production server.

Fix:

This issue can appear for several reasons. The most common is that the application or webserver error message discloses the IP address. This can be solved by determining where to turn off detailed error messages in the application or webserver. Another common reason is due to a comment located in the source of the webpage. This can easily be removed from the source of the page.

File Names:

- http://10.100.92.99:80/frontend/runtime/debug/54e155c810dc9.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b8469e0b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b69768b7.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ff6fc66.data
- http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx
- http://10.100.92.99:80/frontend/runtime/debug/54e066fec56e2.data
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=4f44493
- http://10.100.92.99:80/frontend/runtime/debug/54e14f6f32f33.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b6c1e01c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d3cdf9e.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ca75c8f7.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06cab7eb87.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce1a82ee.data
- http://10.100.92.99:80/frontend/runtime/logs/app.log.1
- http://10.100.92.99:80/frontend/runtime/debug/54e069438de91.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f0e1995b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f1a428d9.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155c7da00c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155851deaf.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06954e6e7d.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f0833ad2.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06cb778a42.data
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=4f44493d&
- http://10.100.92.99:80/frontend/runtime/debug/54e066feeb3f3.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ffa08d0.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d3b555d.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d40a509.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b699d6f8.data
- http://10.100.92.99:80/frontend/runtime/debug/54e186c3455c0.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066e6c249f.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068cad2529.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d361d20.data
- http://10.100.92.99:80/frontend/runtime/debug/54e069434976f.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b83e1d23.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d3b519c.data

- http://10.100.92.99:80/frontend/runtime/debug/54e155cc17a09.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f0ddaee6.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f08daa5f.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f08778a5.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b847a78b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b6951238.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b69d45ac.data
- http://10.100.92.99:80/frontend/runtime/logs/app.log
- http://10.100.92.99:80/frontend/runtime/debug/54e0637656735.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce15ac44.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ff4af4c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f1a8046d.data
- http://10.100.92.99:80/frontend/runtime/debug/54e186c0d216f.data
- http://10.100.92.99:80/frontend/runtime/debug/index.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce1d77f1.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ea9220d.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b841634b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b84b4c98.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce1803ce.data
- http://10.100.92.99:80/frontend/web/student/activity/training-student-status.aspx
- http://10.100.92.99:80/frontend/runtime/debug/54e063c38dcc1.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06343a7091.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066e73251c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ff4b217.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068ca8b2ac.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b846c523.data
- http://10.100.92.99:80/frontend/web/student/student/profile.aspx

Low

Poor Error Handling: Unhandled Exception

Summary:

A server error message was detected. Certain conditions, such as an application failure, will cause a server error message to be displayed. While error messages in and of themselves are not dangerous, per se, it is what an attacker can glean from them that might cause eventual problems. Recommendations include designing and adding consistent error handling mechanisms which are capable of handling any user input to your web application, providing meaningful detail to end-users, and preventing error messages that might provide information useful to an attacker from being displayed.

Implication:

The page body contained an error message. Knowing whether certain inputs trigger a server error can aid or inform an attacker of potential vulnerabilities.

Fix:

For Security Operations:

Information on configuring PHP error messages can be found here: http://us.php.net/manual/en/ref.errorfunc.php.

Use these general recommendations when configuring error messages for display. Also be advised that unknown application testing seeks to uncover new vulnerabilities in both custom and commercial software. Please note that this vulnerability may be a false positive if the page it is flagged on is technical documentation.

- **Use Uniform Error Codes:** Ensure that you are not inadvertently supplying information to an attacker via the use of inconsistent or "conflicting" error messages. For instance, don't reveal unintended information by using error messages

such as Access Denied, which will also let an attacker know that the file he seeks actually exists. Use consistent terminology for files and folders that do exist, do not exist, and which have read access denied.
- **Informational Error Messages:** Ensure that error messages do not reveal too much information. Complete or partial paths, variable and file names, row and column names in tables, and specific database errors should never be revealed to the end user. Remember, an attacker will gather as much information as possible, and then add pieces of seemingly innocuous information together to craft an attack.
- **Proper Error Handling:** Use generic error pages and error handling logic to inform end users of potential problems. Do not provide system information or other data that could be used by an attacker when orchestrating an attack.

For Development:

This problem arises from the improper validation of characters that are accepted by the application. Any time a parameter is passed into a dynamically-generated web page, you must assume that the data could be incorrectly formatted. The application should contain sufficient logic to handle any situation in which a parameter is not being passed or is being passed incorrectly. Keep in mind how the data is being submitted, as a result of a GET or a POST. Additionally, to develop secure and stable code, treat cookies the same as parameters. The following recommendations will help ensure that you are delivering secure web applications.

- **Stringently define the data type:** Stringently define the data type (a string, an alphanumeric character, etc.) that the application will accept. Validate input for improper characters. Adopt the philosophy of using what is good rather than what is bad. Define the allowed set of characters. For instance, if a field is to receive a number, allow that field to accept only numbers. Define the maximum and minimum data lengths that the application will accept.
- **Verify parameter is being passed:** If a parameter that is expected to be passed to a dynamic Web page is omitted, the application should provide an acceptable error message to the user. Also, never use a parameter until you have verified that it has been passed into the application.
- **Verify correct format:** Never assume that a parameter is of a valid format. This is especially true if the parameter is being passed to a SQL database. Any string that is passed directly to a database without first being checked for proper format can be a major security risk. Also, just because a parameter is normally provided by a combo box or hidden field, do not assume the format is correct. A hacker will first try to alter these parameters while attempting to break into your site.
- **Verify file names being passed in via a parameter:** If a parameter is being used to determine which file to process, never use the file name before it is verified as valid. Specifically, test for the existence of characters that indicate directory traversal, such as .../, c:\, and /.
- **Do not store critical data in hidden parameters:** Many programmers make the mistake of storing critical data in a hidden parameter or cookie. They assume that since the user doesn't see it, it's a good place to store data such as price, order number, etc. Both hidden parameters and cookies can be manipulated and returned to the server, so never assume the client returned what you sent via a hidden parameter or cookie.

For QA:

From a testing perspective, ensure that the error handling scheme is consistent and does not reveal private information about your web application. A seemingly innocuous piece of information can provide an attacker the means to discover additional information that can be used to conduct an attack. Make the following observations:

- Do you receive the same type of error for existing and non-existing files?
- Does the error include phrases (such as "Permission Denied") that could reveal the existence of a file?

Reference:

Apache:
Security Tips for Server Configuration
Protecting Confidential Documents at Your Site
Securing Apache - Access Control

IIS:
Implementing NTFS Standard Permissions on Your Web Site

General:
Password-protecting web pages
Web Security

File Names:

- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=4f44493
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=http://ww

- http://10.100.92.99:80/frontend/web/student/student/profile.aspx
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=4f44493d&
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=4f44493d&
- http://10.100.92.99:80/frontend/web/file/download.aspx?file=%00
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=4f44493
- http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx?training_id=h
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx
- http://10.100.92.99:80/frontend/web/student/student/profile.aspx
- http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx
- http://10.100.92.99:80/frontend/web/student/student/profile.aspx
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=http://
- http://10.100.92.99:80/frontend/web/student/student/profile.aspx
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=4f44493
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=4f44493d&

Low

Access Control: Unprotected File

Summary:

A documentation file was found. The danger in having a documentation file available is that it reveals to attackers what type of software you are using and often the specific version information, or a location from where the attacker could download the software itself. Recommendations include removing this file from the production server.

Execution:

Open a web browser and navigate to http://10.100.92.99:80/frontend/web/assets/8969b36/license.txt.

Implication:

The disclosed documentation may aid an attacker in attacking the server and application.

Fix:

For Security Operations:
Remove documentation files from all web accessible locations, or restrict access to the files via access control mechanisms.

For Development:
Have Security Operations remove this file from the production server.

For QA:
Have Security Operations remove this file from the production server.

File Names:

- http://10.100.92.99:80/frontend/web/assets/8969b36/license.txt

Low

Poor Error Handling: Server Error Message

Summary:

A server error response was detected. The server could be experiencing errors due to a misbehaving application, a misconfiguration, or a malicious value sent during the auditing process. While error responses in and of themselves are not dangerous, per se, the error responses give attackers insight into how the application handles error conditions. Errors that can be remotely triggered by an attacker can also potentially lead to a denial of service attack or other more severe vulnerability. Recommendations include designing and adding consistent error handling mechanisms which are capable of handling any user input to your web application, providing meaningful detail to end-users, and preventing error messages that might provide information useful to an attacker from being displayed.

Implication:

The server has issued a 500 error response. While the body content of the error page may not expose any information about the technical error, the fact that an error occurred is confirmed by the 500 status code. Knowing whether certain inputs trigger a server error can aid or inform an attacker of potential vulnerabilities.

Fix:

For Security Operations:

Server error messages, such as "File Protected Against Access", often reveal more information than intended. For instance, an attacker who receives this message can be relatively certain that file exists, which might give him the information he needs to pursue other leads, or to perform an actual exploit. The following recommendations will help to ensure that a potential attacker is not deriving valuable information from any server error message that is presented.

- Uniform Error Codes: Ensure that you are not inadvertently supplying information to an attacker via the use of inconsistent or "conflicting" error messages. For instance, don't reveal unintended information by utilizing error messages such as Access Denied, which will also let an attacker know that the file he seeks actually exists. Have consistent terminology for files and folders that do exist, do not exist, and which have read access denied.
- Informational Error Messages: Ensure that error messages do not reveal too much information. Complete or partial paths, variable and file names, row and column names in tables, and specific database errors should never be revealed to the end user. Remember, an attacker will gather as much information as possible, and then add pieces of seemingly innocuous information together to craft a method of attack.
- Proper Error Handling: Utilize generic error pages and error handling logic to inform end users of potential problems. Do not provide system information or other data that could be utilized by an attacker when orchestrating an attack.

Removing Detailed Error Messages

Find instructions for turning off detailed error messaging in IIS at this link:

http://support.microsoft.com/kb/294807

For Development:

From a development perspective, the best method of preventing problems from arising from server error messages is to adopt secure programming techniques that prevent problems that might arise from an attacker discovering too much information about the architecture and design of your web application. The following recommendations can be used as a basis for that.

- Stringently define the data type (for instance, a string, an alphanumeric character, etc) that the application will accept.
- Use what is good instead of what is bad. Validate input for improper characters.
- Do not display error messages to the end user that provide information (such as table names) that could be utilized in orchestrating an attack.
- Define the allowed set of characters. For instance, if a field is to receive a number, only let that field accept numbers.
- Define the maximum and minimum data lengths for what the application will accept.
- Specify acceptable numeric ranges for input.

For QA:

The best course of action for QA associates to take is to ensure that the error handling scheme is consistent. Do you receive a different type of error for a file that does not exist as opposed to a file that does? Are phrases like "Permission Denied" utilized which could reveal the existence of a file to an attacker? Inconsistent methods of dealing with errors gives an attacker a very powerful way of gathering information about your web application.

Reference:

Apache:
Security Tips for Server Configuration

Protecting Confidential Documents at Your Site
Securing Apache - Access Control

Microsoft:
How to set required NTFS permissions and user rights for an IIS 5.0 Web server
Default permissions and user rights for IIS 6.0
Description of Microsoft Internet Information Services (IIS) 5.0 and 6.0 status codes

File Names:

- http://10.100.92.99:80/frontend/modules/student/views/student/_form.php
- http://10.100.92.99:80/frontend/modules/student/controllers/TrainingClassSubjectTrainerEvaluationCon
- http://10.100.92.99:80/frontend/modules/student/controllers/TrainingScheduleTrainerController.php
- http://10.100.92.99:80/frontend/modules/student/controllers/DefaultController.php
- http://10.100.92.99:80/frontend/models/ContactForm.php
- http://10.100.92.99:80/frontend/models/ResetPasswordForm.php
- http://10.100.92.99:80/frontend/models/TrainingClassSubjectTrainerEvaluation.php
- http://10.100.92.99:80/frontend/modules/student/views/student/password.php
- http://10.100.92.99:80/frontend/modules/student/models/TrainingClassSubjectSearch.php
- http://10.100.92.99:80/frontend/modules/student/models/TrainingScheduleTrainerSearch.php
- http://10.100.92.99:80/frontend/modules/student/controllers/TrainingStudentController.php
- http://10.100.92.99:80/frontend/models/Activity.php
- http://10.100.92.99:80/frontend/models/Satker.php
- http://10.100.92.99:80/frontend/models/TrainingExecutionEvaluation.php
- http://10.100.92.99:80/frontend/views/layouts/main.php
- http://10.100.92.99:80/frontend/modules/student/models/ActivitySearch.php
- http://10.100.92.99:80/frontend/modules/student/models/TrainingClassSubjectTrainerEvaluationSearch.p
- http://10.100.92.99:80/frontend/modules/student/controllers/TrainingController.php
- http://10.100.92.99:80/frontend/modules/student/views/student/view.php
- http://10.100.92.99:80/frontend/modules/student/Module.php
- http://10.100.92.99:80/frontend/web/student/student/profile.aspx
- http://10.100.92.99:80/frontend/controllers/SiteController.php
- http://10.100.92.99:80/frontend/models/ObjectReference.php
- http://10.100.92.99:80/frontend/models/Student.php
- http://10.100.92.99:80/frontend/models/TrainingScheduleTrainer.php
- http://10.100.92.99:80/frontend/modules/student/controllers/StudentController.php
- http://10.100.92.99:80/frontend/modules/student/views/training-execution-evaluation/
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=4f44493
- http://10.100.92.99:80/frontend/modules/student/views/student/profile.php
- http://10.100.92.99:80/frontend/web/file/download.aspx?file=%00
- http://10.100.92.99:80/frontend/web/student/activity/training-student-status.aspx
- http://10.100.92.99:80/frontend/controllers/FileController.php
- http://10.100.92.99:80/frontend/models/LoginForm.php
- http://10.100.92.99:80/frontend/models/SignupForm.php
- http://10.100.92.99:80/frontend/models/TrainingSchedule.php
- http://10.100.92.99:80/frontend/views/site/
- http://10.100.92.99:80/frontend/modules/student/controllers/ActivityController.php
- http://10.100.92.99:80/frontend/modules/student/controllers/TrainingClassSubjectController.php
- http://10.100.92.99:80/frontend/modules/student/views/training-schedule-trainer/

- http://10.100.92.99:80/frontend/models/Person.php
- http://10.100.92.99:80/frontend/models/TrainingClass.php
- http://10.100.92.99:80/frontend/modules/student/views/activity/
- http://10.100.92.99:80/frontend/modules/student/views/training-class-subject-trainer-evaluation/
- http://10.100.92.99:80/frontend/modules/student/models/TrainingSearch.php
- http://10.100.92.99:80/frontend/models/PasswordResetRequestForm.php
- http://10.100.92.99:80/frontend/models/Training.php
- http://10.100.92.99:80/frontend/models/TrainingStudent.php
- http://10.100.92.99:80/frontend/modules/student/views/default/
- http://10.100.92.99:80/frontend/modules/student/models/TrainingExecutionEvaluationSearch.php
- http://10.100.92.99:80/frontend/modules/student/controllers/TrainingExecutionEvaluationController.ph
- http://10.100.92.99:80/frontend/models/ProgramSubject.php
- http://10.100.92.99:80/frontend/models/TrainingClassStudent.php
- http://10.100.92.99:80/frontend/widgets/Alert.php
- http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx
- http://10.100.92.99:80/frontend/modules/student/models/TrainingClassStudentSearch.php
- http://10.100.92.99:80/frontend/modules/student/views/training-class-subject/
- http://10.100.92.99:80/frontend/modules/student/models/TrainingStudentSearch.php
- http://10.100.92.99:80/frontend/models/ActivitySearch.php
- http://10.100.92.99:80/frontend/models/Reference.php
- http://10.100.92.99:80/frontend/models/TrainingClassSubject.php
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=4f44493d&

Low

Cross-Site Scripting: Charset Control

Summary:

A vulnerability was detected in your web application that allows a user to control the HTML character encoding used to parse the HTTP response of a given request. Attackers can exploit this vulnerability to evade certain validation mechanisms used for Cross-site Scripting.
The response character encoding is used by a web browser to decide how to interpret the characters in the body of the HTTP response. The most common encoding used by web applications today is UTF-8. The character set (charset) declaration is usually done through a header in the HTTP response or using the HTML <meta> tag. Such declarations should be controlled by the application only. If this declaration is controlled through user input, then an attacker can use this feature to modify the charset that will be used by the browser and modify the interpretation of the contents of the response. This can allow for Cross-site Scripting attacks that would otherwise not have succeeded while using UTF-8 encoding.

Execution:

This vulnerability is detected by modifying an input parameter value in an HTTP request to a different charset. The headers and the HTML <meta> tag of the corresponding HTTP response are then observed to confirm a successful manipulation of the response charset.

Implication:

A successful exploitation of this vulnerability could increase the probability of a successful Cross-site Scripting attack by evading existing server-side input validation routines.
For example:

```
+ADw-script+AD4-alert(document.location)+ADw-/script+AD4
```

The above string means nothing in most encoding types, and therefore is "safe", but when a victim views this under utf-7 encoding, it will be interpreted as valid html tag and hence, the script will be executed.

Fix:

The ideal solution would be to control charset declarations from the application and never through user-supplied input. If a particular feature of the application demands such a capability, then it is advisable to use a white list of allowed charsets and validate that proper input validation routines are in place for all the values in the white list.

Reference:

Browser Security Handbook
Computer Security Research - Secunia
Maia Mailguard 'charset' Parameter HTML Injection Vulnerability
OWASP Encoding Project

File Names:

- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?ActivitySearch%5Bname%5D=12345&Activ
- http://10.100.92.99:80/frontend/web/student/default/index.aspx?ActivitySearch%5Bname%5D=12345&Activi
- http://10.100.92.99:80/frontend/web/student/default/index.aspx?satker_id=&year=2015&_pjax=%23pjax-gr
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?satker_id=&year=2015&_pjax=%23pjax-g

Low

Server Misconfiguration: Response Headers

Summary:

Missing a Content-Type header in the HTTP Response could expose the application to Cross-Site Scripting vulnerabilities via:

Content Sniffing Mismatch
Failure to explicitly specify the type of the content served by the requested resource can allow attackers to conduct Cross-Site Scripting attacks by exploiting the inconsistencies in content sniffing techniques employed by the browsers.
The Content-Type header is used by:

- The web server to dictate how the requested resource is interpreted by the user agent. In the absence of this header the browser depends on content sniffing algorithms to guess the type of content and render or interpret it accordingly.
- File upload filters to discard file types not allowed by the application. In the absence of a Content-Type header, the file upload filter relies on the file extension or the content of the file to detect and store an appropriate mime type for the uploaded file.

The lack of explicit content type specification can allow attackers to exploit the mismatch between the mime sniffing algorithm used by the browser and upload filter. By uploading files with benign extensions (like .jpg), an attacker can easily bypass the upload filter to upload files containing malicious HTML content. The browser's content sniffing algorithm will however render it as HTML based on the content of the file thus executing any malicious scripts embedded within the HTML content.

Character Set Mismatch

Character set specification is part of the Content-Type header. Absence of this specification could allow attackers to bypass input validation filters or HTML entity escape functionality and conduct Cross-Site Scripting attacks against the target application. When the character set is not specified, browsers will attempt to guess the most appropriate character set. This could result in a mismatch between the character set assumed by the application during the generation of the content and by the browser during the parsing and interpretation of the same content. An attacker can exploit this inconsistency to encode attacks using a character set that'll hide the malicious payloads from the valdiation filters and escaping mechanisms put in place by the application but at the same time will be interpreted by the browser as a valid executable entity.

Execution:

Below example scenarios demonstrate the exploitation of the weakness:

Content Sniffing Mismatch

. Attacker uploads a file with .jpg extension and no Content-Type specification. The file contains malicious HTML and JavaScript content embedded inside.

. In the absence of the Content-Type header, the application saves the uploaded file along with the mime type of the .jpg

. The attacker uses social engineering to entice the desired target into accessing the uploaded file

. Upon receiving the requested file without the Content-Type header, the target's browser assumes the content type to be HTML based on the HTML and JavaScript content inside and renders the file causing attacker's JavaScript payload to be executed.

Character Set Mismatch

0. Attacker converts the desired payload of <script>alert(document.location)</script> into UTF-7 encoded string +ADw-script+AD4-alert(document.location)+ADw-/script+AD4 and sends it to the vulnerable application.

. An application using the ISO-8859-1 character set for filtering or escaping special characters will fail to detect the the '<' and '>' characters as dangerous

. The absence of character set specification due to the missing Content-Type header will force the browser to guess the character set to use for rendering the application response containing the attacker's payload. If the browser correctly guesses the encoding as UTF-7, the injected payload will be successfully executed.

Implication:

The application fails to impose constraints on the parsing and interpretation of the response content; allowing attackers to bypass validation filters or escaping functionality and introduce malicious scripts and force the browser to execute the desired payload.

Fix:

Configure the server to send the appropriate content type and character set information for the requested resource.

Reference:

Server Configuration
Mime Types in IIS 7
Content Negotiation - Apache HTTP Server

Content Sniffing:
Mime Sniffing Standard
Content Sniffing Signatures
Secure Content Sniffing for Web Browsers [PDF]

OWASP:
OWASP Testing Guide Appendix D: Encoded Injection

File Names:

- http://10.100.92.99:80/frontend/web/assets/9b922f9d/LICENSE
- http://10.100.92.99:80/frontend/web/assets/b44ebcff/CHANGE.md
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/core.less
- http://10.100.92.99:80/frontend/web/assets/d021eecb/README.md
- http://10.100.92.99:80/frontend/web/assets/b604c748/_config.yml
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/aliens.erb
- http://10.100.92.99:80/frontend/web/assets/b604c748/Gemfile.lock
- http://10.100.92.99:80/frontend/web/assets/d021eecb/LICENSE
- http://10.100.92.99:80/frontend/web/assets/9b922f9d/README.md
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_bordered-pulled.scss
- http://10.100.92.99:80/frontend/runtime/debug/54e06b6c1e01c.data
- http://10.100.92.99:80/frontend/web/assets/189fe31c/jquery.min.map
- http://10.100.92.99:80/frontend/runtime/logs/app.log.1
- http://10.100.92.99:80/frontend/web/assets/b604c748/CONTRIBUTING.md
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/run-qunit.coffee
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/LICENSE
- http://10.100.92.99:80/frontend/web/assets/ea5317b3/CHANGE.md

- http://10.100.92.99:80/frontend/web/assets/b604c748/Gemfile
- http://10.100.92.99:80/frontend/runtime/logs/app.log
- http://10.100.92.99:80/frontend/web/assets/b604c748/fonts/FontAwesome.otf
- http://10.100.92.99:80/frontend/web/assets/ace9a976/fonts/glyphicons-halflings-regular.ttf
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/CHANGELOG.md
- http://10.100.92.99:80/frontend/web/assets/8969b36/README.rdoc

Informational **System Information Leak: Filename Found in Comments**

Summary:

A URL or filename was found in the comments of the file.

File Names:

- http://10.100.92.99:80/frontend/web/student/activity/index.aspx
- http://10.100.92.99:80/frontend/web/student/activity/view.aspx?id=38
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?sort=name
- http://10.100.92.99:80/frontend/web/student/activity/class-student.aspx?training_id=82&class_id=38&p
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?ActivitySearch%5Bname%5D=12345&Activ
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?satker_id=&year=2015&_pjax=%23pjax-g
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?satker_id=&year=2015&_pjax=%23pjax-g
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?satker_id=&year=2015&_pjax=%23pjax-g
- http://10.100.92.99:80/frontend/web/assets/16926e51/js/bootstrap-datepicker.js
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?ActivitySearch%5bname%5d=12345&Activ
- http://10.100.92.99:80/frontend/web/student/activity/class-student.aspx?training_id=82&class_id=38
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?ActivitySearch%5Bname%5D=12345&Activ
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?ActivitySearch%5Bname%5D=12345&Activ

Informational **Hidden Field**

Summary:

While preventing display of information on the web page itself, the information submitted via hidden form fields is easily accessible, and could give an attacker valuable information that would prove helpful in escalating his attack methodology. Recommendations include not relying on hidden form fields as a security solution for any area of the web application that contains sensitive information or access to privileged functionality such as remote site administration functionality.

Execution:

Any attacker could bypass a hidden form field security solution by viewing the source code of that particular page.

Implication:

The greatest danger from exploitation of a hidden form field design vulnerability is that the attacker will gain information that will help in orchestrating a far more dangerous attack.

Fix:

Do not rely on hidden form fields as a method of passing sensitive information or maintaining session state. One workable bypass is to encrypt the hidden values in a form, and then decrypt them when that information is to be utilized by a database operation or a script. From a security standpoint, the best method of temporarily storing information required by different forms is to utilize a session cookie.

Whether hidden or not, if your site utilizes values submitted via a form to construct database queries, do not make the assumption that the data is non-malicious. Instead, utilize the following recommendations to sanitize user supplied input.

- Stringently define the data type (for instance, a string, an alphanumeric character, etc) that the application will accept.

- Use what is good instead of what is bad.
- Validate input for improper characters.
- Do not display error messages to the end user that provide information (such as table names) that could be utilized in orchestrating an attack.
- Define the allowed set of characters. For instance, if a field is to receive a number, only let that field accept numbers.
- Define the maximum and minimum data lengths for what the application will accept.
- Specify acceptable numeric ranges for input.

File Names:

- http://10.100.92.99:80/frontend/web/student/activity/class.aspx?training_id=4f44493d&training_studen
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?satker_id=&year=2015&_pjax=%23pjax-g
- http://10.100.92.99:80/frontend/web/student/student/view.aspx
- http://10.100.92.99:80/frontend/web/student/student/profile.aspx
- http://10.100.92.99:80/frontend/web/student/activity/student.aspx?training_id=4f44493d&training_stud
- http://10.100.92.99:80/frontend/web/student/default.aspx
- http://10.100.92.99:80/frontend/web/site/masuk.aspx
- http://10.100.92.99:80/frontend/web/student/activity/create.aspx
- http://10.100.92.99:80/frontend/web/student/activity/class-student.aspx?training_id=82&class_id=38
- http://10.100.92.99:80/frontend/web/student/student/password.aspx
- http://10.100.92.99:80/frontend/web/student/activity/update.aspx?id=38
- http://10.100.92.99:80/frontend/web/student/default/index.aspx?satker_id=&year=2015&_pjax=%23pjax-gr
- http://10.100.92.99:80/frontend/web/student/activity/training-student-status.aspx?status=4d673d3d&tr
- http://10.100.92.99:80/frontend/web/site/request-password-reset.aspx
- http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx?training_id=4

Informational

System Information Leak: OPTIONS HTTP Method

Summary:

The server supports the OPTIONS HTTP method. The OPTIONS method is used to determine what other methods the server supports for a given URI/resource.

Reference:

RFC 2616 Section 9: HTTP Methods:
http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html

Apache:
Apache HTTP Server Version 2.0
Apache HTTP Server Version 1.3

Microsoft:
UrlScan Security Tool
How to configure the URLScan Tool
Setting Application Mappings in IIS 6.0

File Names:

- http://10.100.92.99:80/

Informational **Access Control: Unprotected File**

Summary:

A Flash movie or Flash object was found. Flash movies and objects can be decompiled and may contain sensitive information. An attacker could decompile the Flash file and gain access to the confidential information, including any hard-coded passwords and keys, within the Flash file.

Execution:

A primary tool in the arsenal of the attacker who wants to get inside your code is the decompiler. A decompiler takes an executable file and attempts to re-create the original source code. It may be almost impossible to go from machine code to a high-level language. It is, however, easy to recover an assembly language version of the program.

Implication:

The attacker’s goal in re-creating the original source code may include one or more of the following:

- To steal a valuable algorithm for use in his own code
- To understand how a security function works to enable him to bypass it
- To extract confidential information, such as hard-coded passwords and keys
- To enable him to alter the code so that it behaves in a malicious way

Reference:

Flare - Flash Decompiler
http://www.nowrap.de/flare.html

File Names:

- http://10.100.92.99:80/frontend/web/assets/b44ebcff/README.md
- http://10.100.92.99:80/frontend/web/assets/8969b36/script/soundmanager2-nodebug.js
- http://10.100.92.99:80/frontend/web/assets/b44ebcff/js/fileinput.js

Informational **Poor Error Handling: Unhandled Exception**

Summary:

Error messages related to opening files were detected. These errors may reveal include file names, file system paths or application execution details. Recommendations include resolving the program errors or restricting access to the programs effected.

Execution:

Click http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx to verify the information in a web browser.

Implication:

Details of the application or system may be revealed through the error messages.

Fix:

Resolve the file open errors through a program or permissions change. If necessary, remove the program or restrict access to it to keep the erorrs from being displayed.

File Names:

- http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=4f44493
- http://10.100.92.99:80/frontend/web/file/download.aspx?file=%00
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=4f44493d&

- http://10.100.92.99:80/frontend/web/student/student/profile.aspx

Informational

Session Management: Session Token Discovery

Summary:

The following Session Tokens have been identified in this website:
1. PHPSESSID=lfkmh1bf8sovl2hb4n6vf7gtp0
.

Session tokens play a key role in maintaining state in modern web applications. Conceptually, a session management system contains a collection of state variables that are stored either client- or server-side, and session tokens (also collectively called session identifiers or session IDs) are used as the "key" to access these state variables. Session tokens can be placed in cookies, query/post parameters or other HTTP headers and can be comprised of a single token or referenced in aggregate as a collection of multiple tokens.
Session tokens enable a web application to track an authenticated user's activities, correlate requests sent by that user, and provide appropriate services to the user accordingly. When a user successfully authenticates to a web application, the web application usually associates the user's identity with session tokens and accesses user data by referencing these tokens. Thus, weaknesses in deploying session tokens in a web application can be exploited by malicious attackers to hijack user sessions and compromise data confidentiality, integrity and availability.

Execution:

N/A

Implication:

When session tokens are accessible to malicious attackers by way of a vulnerable implementation, they can break into the corresponding login sessions and steal or tamper with sensitive user data. In particular, if these session tokens are tied to an administrative account, the whole website, including user accounts and the data stored within, is at serious risk of compromise.

Fix:

N/A

Reference:

Session Management:
http://msdn.microsoft.com/en-us/library/aa478989.aspx

ASP.NET Session State Overview:
http://msdn.microsoft.com/en-us/library/ms178581(v=vs.100).aspx

Maintaining Client State using Java Servlet Technology:
http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Servlets11.html

PHP Sessions
http://php.net/manual/en/features.sessions.php

OWASP Session Management Cheat Sheet:
https://www.owasp.org/index.php/Session_Management_Cheat_Sheet

File Names:

- http://10.100.92.99:80/frontend/web/site/masuk.aspx

Best Practice

Compliance Failure: Missing Privacy Policy

Summary:

A privacy policy was not supplied by the web application within the scope of this audit. Many legislative initiatives require that organizations place a publicly accessible document within their web application that defines their website's privacy policy. As a general rule, these privacy policies must detail what information an organization collects, the purpose for collecting it, potential avenues of disclosure, and methods for addressing potential grievances.
Various laws governing privacy policies include the Gramm-Leach-Bliley Act, Health Insurance Portability and Accountability Act (HIPAA), the California Online Privacy Protection Act of 2003, European Union's Data Protection Directive and others.

Execution:

All of the web pages accessible within the scope of the scan are sampled for textual content that often constitutes a privacy policy statement. A violation is reported upon completion of the web application crawl without a successful match against any of the web pages.
Note that the privacy policy of your application could be located on another host or within a section of the site that was not configured as part of the scan. To validate, please try to access the privacy policy of your website and check to see if it was part of the scan.

Implication:

Most privacy laws are created to protect residents who are users of the website. Hence, organizations from any part of the world must adhere to these laws if they cater to customers residing in these geographical areas. Failing to do so could result in a lawsuit by the corresponding government against the organization.

Fix:

Declare a comprehensive privacy policy for the website, and ensure that it is accessible from every page that seeks personal information from users. To verify the fix, rescan the site in order to discover and audit the newly added resources.

Descriptions:
Any standard web application privacy policy should include the following components:

- A description of the intended purpose for collecting the data.
- A description of the use of the data.
- Methods for limiting the use and disclosure of the information.
- A list of the types of third parties to whom the information might be disclosed.
- Contact information for inquires and complaints.

Reference:

California Online Privacy Protection Act
http://oag.ca.gov/privacy/COPPA

National Conference of State Legislation
http://www.ncsl.org/issues-research/telecom/state-laws-related-to-internet-privacy.aspx

Gramm-Leach-Bliley Act
http://www.gpo.gov/fdsys/pkg/PLAW-106publ102/pdf/PLAW-106publ102.pdf

Health Insurance Portability and Accountability Act of 1996
https://www.cms.gov/Regulations-and-Guidance/HIPAA-Administrative-Simplification/HIPAAGenInfo/downloads/HIPAALaw.pdf

Health Insurance Portability and Accountability Act of 1996
http://ec.europa.eu/justice/policies/privacy/docs/guide/guide-ukingdom_en.pdf

File Names:
- http://10.100.92.99:80/frontend/web/

Best Practice

Privacy Violation: Autocomplete

Summary:

Most recent browsers have features that will save form field content entered by users and then automatically complete form entry the next time the fields are encountered. This feature is enabled by default and could leak sensitive information since it is stored on the hard drive of the user. The risk of this issue is greatly increased if users are accessing the application from a shared environment. Recommendations include setting autocomplete to "off" on all your forms.

Reference:

Microsoft:
Autocomplete Security

File Names:
- http://10.100.92.99:80/frontend/web/student/student/view.aspx
- http://10.100.92.99:80/frontend/web/site/request-password-reset.aspx
- http://10.100.92.99:80/frontend/web/student/default.aspx
- http://10.100.92.99:80/frontend/web/student/activity/create.aspx
- http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx?training_id=4
- http://10.100.92.99:80/frontend/web/student/activity/class.aspx?training_id=4f44493d&training_studen
- http://10.100.92.99:80/frontend/web/site/masuk.aspx
- http://10.100.92.99:80/frontend/web/assets/ea5317b3/examples/
- http://10.100.92.99:80/frontend/web/student/activity/class-student.aspx?training_id=82&class_id=38
- http://10.100.92.99:80/frontend/web/student/activity/index.aspx?satker_id=&year=2015&_pjax=%

23pjax-g

- http://10.100.92.99:80/frontend/web/student/student/profile.aspx
- http://10.100.92.99:80/frontend/web/student/default/index.aspx?satker_id=&year=2015&_pjax=%23pjax-gr
- http://10.100.92.99:80/frontend/web/student/activity/update.aspx?id=38
- http://10.100.92.99:80/frontend/web/student/activity/student.aspx?training_id=4f44493d&training_stud

Best Practice

Weak Cryptographic Hash

Summary:

A string of hexadecimal digits matching the length of a cryptographic hash from the MD family was detected. Cryptographic hashes are often used to protect passwords, session information, and other sensitive data. There are multiple hashing algorithms in the MD family. By far the most commonly used algorithm is MD5, though MD4 and MD2 are still used with various public key and digital certificate systems. There are known attacks against MD5, MD4, and MD2. These hashes are also susceptible to Rainbow table attacks unless the input is properly salted. As such the MD family of cryptographic hashing functions should not be considered secure and should only be used in certain situations.

Implication:

Hashes produced by the MD family should only be used for short-lived uses where the hash and/or hashed data is not highly security sensitive, or for uses where uniqueness is not a critical requirement. MD Hashes should not be used for any type of long term application such as verifying the integrity of a file or for password storage.

Fix:

For Development:
The application should only use cryptographically secure hashing algorithms, such as SHA-224, SHA-256, SHA-384, or SHA-512. Hashes representing sensitive data should be salted to reduce the effectiveness of rainbow tables.

For Security Operations:
Implement a security policy that precludes the use of MD5, MD4, or MD2 for cryptographic functionality.

For QA:
Make sure that the application is not relying on MD5, MD4, or MD2 for cryptographic functionality.

Reference:

MD5
http://en.wikipedia.org/wiki/MD5
Cryptographic Salting
http://en.wikipedia.org/wiki/Salt_%28cryptography%29
Project Rainbow Crack
http://www.antsight.com/zsl/rainbowcrack/

File Names:

- http://10.100.92.99:80/frontend/runtime/cache/96/
- http://10.100.92.99:80/frontend/web/student/training-execution-evaluation.aspx?training_id=4f44493d&
- http://10.100.92.99:80/frontend/web/file/download.aspx?file=%00
- http://10.100.92.99:80/frontend/web/student/activity/training-student-status.aspx
- http://10.100.92.99:80/frontend/web/student/training-schedule-trainer/index.aspx?training_id=4f44493
- http://10.100.92.99:80/frontend/web/student/activity/view-training-student-status.aspx
- http://10.100.92.99:80/frontend/web/student/student/profile.aspx

Best Practice

Flash Bad Practices: Embedded SWF Settings

Summary:

Examination of the ActionScript revealed that the application is using both the browser and the network communication APIs. When embedding this SWF in an HTML page one should set the AllowNetworkingAccess flag to “all”.

When a SWF is embedded within HTML, there are several flags which inform the Flash player if the SWF file should have access to content from the browser or from the network. The AllowNetworkingAccess flag tells the Flash player to disallow the SWF from communicating with the browser and HTML DOM using ExternalInterface, fscommand or getURL. The AllowNetworkingAccess flag informs the Flash player that is it allowed to make networking calls like XML.load, loadVariables, LoadVars.load etc. If a Flash application needs to communicate with both the browser and the network, the AllowNetworkingAccess tag should be set to "all".

Implication:

In case of a successful attack on a SWF application, an attacker could gain control over the application and send unauthorized information to an arbitrary site.

Fix:

When embedding the SWF file in an HTML page use the allowScriptAccess and allowNetworking parameters to restrict the browser and remote communication abilities of SWF applications.
Inside the HTML page, embed the SWF in the following manner

```
<object id='MyMovie.swf' classid='clsid:D27CDB6E-AE6D-11cf-96B8-444553540000'
codebase='http://download.adobe.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,0,0' height='100%'
width='100%'><param name='allowNetworking' value='all'/><param name='allowScriptAccess'
value='sameDomain'/><param name='src' value=''MyMovie.swf'/><embed name='MyMovie.swf'
pluginspage='/go/getflashplayer' src='MyMovie.swf' height='100%' width='100%' allowNetworking='none'/></object>
```

Reference:

Adobe:
Creating more secure SWF web applications

Restricting SWF content from HTML

OWASP:
OWASP Flash Security Project

File Names:

- http://10.100.92.99:80/frontend/web/assets/8969b36/swf/soundmanager2.swf
- http://10.100.92.99:80/frontend/web/assets/8969b36/swf/soundmanager2_debug.swf

Best Practice

Session Management: Easy-to-Guess Session Identifier Name

Summary:

A well-known session cookie has been identified: PHPSESSID .</drc_knownsessiontokens>
Many web application frameworks, such as ASP.NET, implement their own session management solution based on HTTP cookies and they provide APIs that developers can use to integrate their session-dependent functionalities with these built-in solutions. Each web application development framework has its well-known default session ID, e.g., ASP.NET_Sessionid for ASP.Net. If these default session IDs are used in web applications, their purpose and underlying technologies are revealed to attackers immediately. One best practice in securing session management is to anonymize or otherwise obscure well-known session IDs.

Execution:

An attacker can collect a list of well-known session IDs used in web application frameworks and search for the target session ID from the list.

Implication:

Although disclosing the information that a certain cookie is used as session ID and what server technologies are used in a web site doesn't directly result in the compromise of the web site, they do facilitate any further attackers a malicious intruder may be attempting.

Fix:

Developers should customize the session cookies by following the instructions provided by the web application framework on which their application is running. For example, to change the default name of a session cookie in ASP.NET, the developer can simply modify the attribute of sessionState tag in the application setting file (web.config) as shown below:

```
<configuration>
   <system.web>
      <sessionstate cookiename="new name">
      </sessionstate>
   </system.web>
</configuration>
```

Reference:

OWASP Session Management Cheat Sheet
https://www.owasp.org/index.php/Session_Management_Cheat_Sheet

OWASP Cookies Database
https://www.owasp.org/index.php/Category:OWASP_Cookies_Database

SessionSate Element (ASP.NET Settings Schema)
http://msdn.microsoft.com/en-us/library/h6bb9cz9(v=vs.100).aspx

File Names:

- http://10.100.92.99:80/frontend/web/site/masuk.aspx

Best Practice

Server Misconfiguration: Response Headers

Summary:

The Content-Type HTTP response header or the HTML meta tag provides a mechanism for the server to specify an appropriate character encoding for the response content to be rendered in the web browser. Proper specification of the character encoding through the charset parameter in the Content-Type field reduces the likelihood of misinterpretation of the characters in the response content and ensure reliable rendering of the web page.Failure to ensure enforcement of the desired character encoding could result in client-side attacks like Cross-Site Scripting.

Execution:

Verify the character set specification on every HTTP response. Character sets can be specified in the HTTP header or in an HTML meta tag. In the case of an XML response, the character set can be specified along with the XML Declaration.

Implication:

In the absence of the character set specification, a user-agent might default to a non-standard character set, or could derive an incorrect character set based on certain characters in the response content. In some cases, both these approaches can cause the response to be incorrectly rendered. This may enable other attacks such as Cross-site Scripting.

Fix:

Ensure that a suitable character set is specified for every response generated by the web application. This can be done either by,

- Modifying the code of the web application, which would require all pages to be modified.
- Adding Content-Type header to the server configuration (**recommended**). This ensures that the header is added to all the responses with minimal development effort.

Reference:

DoD Application Security and Development STIG
http://iase.disa.mil/stigs/app_security/app_sec/app_sec.html

UTF-7 encoding used to create XSS attack
http://www.securityfocus.com/archive/1/420001

File Names:

- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/hello.erb
- http://10.100.92.99:80/frontend/web/assets/9b922f9d/README.md
- http://10.100.92.99:80/frontend/runtime/debug/54e06b847a78b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06cab7eb87.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066e6c249f.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ffa08d0.data
- http://10.100.92.99:80/frontend/runtime/debug/54e069438de91.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f1a428d9.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155851deaf.data
- http://10.100.92.99:80/frontend/web/assets/ea5317b3/composer.json
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/font-awesome.less
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/path.less
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/variables.less
- http://10.100.92.99:80/frontend/web/assets/b604c748/src/
- http://10.100.92.99:80/frontend/web/assets/d021eecb/component.json
- http://10.100.92.99:80/frontend/web/assets/b44ebcff/README.md
- http://10.100.92.99:80/frontend/web/assets/b604c748/Gemfile.lock
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_extras.scss
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_list.scss

- http://10.100.92.99:80/frontend/web/assets/d021eecb/package.json
- http://10.100.92.99:80/frontend/web/assets/b44ebcff/bower.json
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/aliens.erb
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/home.erb
- http://10.100.92.99:80/frontend/web/assets/9b922f9d/bower.json
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_rotated-flipped.scss
- http://10.100.92.99:80/frontend/runtime/debug/54e06b69d45ac.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b69768b7.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b841634b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066e73251c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d3b519c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06954e6e7d.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f1a8046d.data
- http://10.100.92.99:80/frontend/runtime/debug/index.data
- http://10.100.92.99:80/frontend/web/assets/b604c748/component.json
- http://10.100.92.99:80/frontend/web/assets/189fe31c/jquery.min.map
- http://10.100.92.99:80/frontend/runtime/debug/54e066ea9220d.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d3b555d.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f0ddaee6.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155c7da00c.data
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/LICENSE
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/app.rb
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/boom.erb
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/nested_title.erb
- http://10.100.92.99:80/frontend/web/assets/9b922f9d/composer.json
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_stacked.scss
- http://10.100.92.99:80/frontend/runtime/debug/54e06b83e1d23.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ca75c8f7.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce15ac44.data
- http://10.100.92.99:80/frontend/web/assets/8969b36/README.rdoc
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/bordered-pulled.less
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_fixed-width.scss
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/rotated-flipped.less
- http://10.100.92.99:80/frontend/assets/
- http://10.100.92.99:80/frontend/web/assets/b604c748/composer.json
- http://10.100.92.99:80/frontend/web/assets/d021eecb/release.sh
- http://10.100.92.99:80/frontend/runtime/debug/54e066fec56e2.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068cad2529.data
- http://10.100.92.99:80/frontend/runtime/debug/54e069434976f.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f6f32f33.data
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/CHANGELOG.md
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/spinning.less
- http://10.100.92.99:80/frontend/runtime/debug/54e06b6c1e01c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06b6951238.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce1803ce.data

- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/anchor.erb
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/long.erb
- http://10.100.92.99:80/frontend/web/assets/d021eecb/README.md
- http://10.100.92.99:80/frontend/web/assets/b44ebcff/composer.json
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/core.less
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/larger.less
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_icons.scss
- http://10.100.92.99:80/frontend/web/assets/b604c748/bower.json
- http://10.100.92.99:80/frontend/web/assets/d021eecb/select2.jquery.json
- http://10.100.92.99:80/frontend/web/assets/ea5317b3/LICENSE.md
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/extras.less
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/icons.less
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/dinosaurs.erb
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/referer.erb
- http://10.100.92.99:80/frontend/web/assets/9b922f9d/package.json
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_mixins.scss
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_variables.scss
- http://10.100.92.99:80/frontend/runtime/debug/54e06b84b4c98.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06cb778a42.data
- http://10.100.92.99:80/frontend/runtime/debug/54e063c38dcc1.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ff4af4c.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068ca8b2ac.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f0833ad2.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155cc17a09.data
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/bower.json
- http://10.100.92.99:80/frontend/web/assets/ace9a976/css/bootstrap-theme.css.map
- http://10.100.92.99:80/frontend/web/assets/b604c748/README.md
- http://10.100.92.99:80/frontend/web/assets/b604c748/_config.yml
- http://10.100.92.99:80/frontend/web/assets/d021eecb/bower.json
- http://10.100.92.99:80/frontend/web/assets/ea5317b3/CHANGE.md
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/fixed-width.less
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/list.less
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/stacked.less
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/font-awesome.scss
- http://10.100.92.99:80/frontend/runtime/logs/app.log
- http://10.100.92.99:80/frontend/runtime/logs/app.log.1
- http://10.100.92.99:80/frontend/runtime/debug/54e06b846c523.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ff6fc66.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d3cdf9e.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f08daa5f.data
- http://10.100.92.99:80/frontend/runtime/debug/54e155c810dc9.data
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/composer.json
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/run-qunit.coffee
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/double_title.erb
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/scripts.erb

- http://10.100.92.99:80/frontend/web/assets/9b922f9d/component.json
- http://10.100.92.99:80/frontend/web/assets/9b922f9d/LICENSE
- http://10.100.92.99:80/frontend/web/assets/9b922f9d/select2.jquery.json
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/fragment.erb
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_spinning.scss
- http://10.100.92.99:80/frontend/runtime/debug/54e06b699d6f8.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce1d77f1.data
- http://10.100.92.99:80/frontend/runtime/debug/54e0637656735.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066feeb3f3.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d361d20.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f08778a5.data
- http://10.100.92.99:80/frontend/runtime/debug/54e186c3455c0.data
- http://10.100.92.99:80/frontend/web/assets/b44ebcff/LICENSE.md
- http://10.100.92.99:80/frontend/web/assets/ea5317b3/bower.json
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_core.scss
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_larger.scss
- http://10.100.92.99:80/frontend/web/assets/d021eecb/composer.json
- http://10.100.92.99:80/frontend/web/assets/ace9a976/css/bootstrap.css.map
- http://10.100.92.99:80/frontend/web/assets/b604c748/Gemfile
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/env.erb
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/test/views/timeout.erb
- http://10.100.92.99:80/frontend/web/assets/9b922f9d/release.sh
- http://10.100.92.99:80/frontend/web/assets/d021eecb/LICENSE
- http://10.100.92.99:80/frontend/web/assets/b44ebcff/CHANGE.md
- http://10.100.92.99:80/frontend/web/assets/ea5317b3/README.md
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_bordered-pulled.scss
- http://10.100.92.99:80/frontend/web/assets/b604c748/less/mixins.less
- http://10.100.92.99:80/frontend/web/assets/b604c748/scss/_path.scss
- http://10.100.92.99:80/frontend/runtime/debug/54e06b8469e0b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06ce1a82ee.data
- http://10.100.92.99:80/frontend/runtime/debug/54e06343a7091.data
- http://10.100.92.99:80/frontend/runtime/debug/54e066ff4b217.data
- http://10.100.92.99:80/frontend/runtime/debug/54e068d40a509.data
- http://10.100.92.99:80/frontend/runtime/debug/54e14f0e1995b.data
- http://10.100.92.99:80/frontend/runtime/debug/54e186c0d216f.data
- http://10.100.92.99:80/frontend/web/assets/3d6f93ac/README.md
- http://10.100.92.99:80/frontend/web/assets/b604c748/CONTRIBUTING.md
- http://10.100.92.99:80/frontend/web/assets/b604c748/package.json