Hardening security properties for BPM applications

IBM® Business Process Manager provides configuration settings at the deployment environment level to harden security that mitigates web application threats that include cross-site request forgery (CSRF), network sniffing, clickjacking, and uploading malicious documents.

* These security settings are off by default. You enable them by setting custom properties at the deployment environment (DE) level in the configuration repository by using the **setBPMProperty** command.

Security Hardening Properties

1.ProcessServer.CsrfProtectionRefererWhitelist

* IBM Business Process Manager provides a configuration option to enable CSRF protection, which is off by default. As a result, security testing using tools such as AppScan might identify CSRF vulnerabilities.

* A Cross-Site Request Forgery attack uses HTML and/or Java™Script on one site to trick a user's browser into sending a request to another site. The user's browser will send all cookies applicable to the target site along with the forged request.

* IBM Business Process Manager V7.5.1 Fix Pack 2, V8.0.1 Fix Pack 3, V8.5.0 Fix Pack 1 and V8.5.5.0 introduced a CSRF protection feature that validates the HTTP REFERER header.

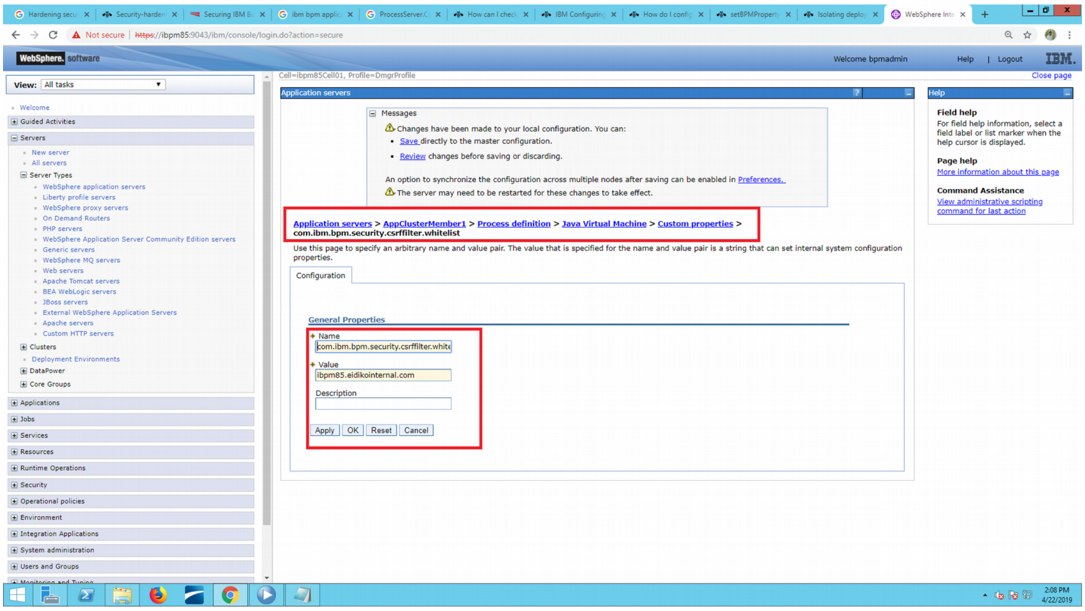
* This feature can be enabled by setting a JVM custom property listing the acceptable values for the HTTP REFERER header. If the header is not empty and contains a value that is no explicitly white listed, the request will be rejected with HTTP 403 (forbidden) and a severe message will be written to the log file: "HTTP request to was blocked by CSRF Filtering"

You need to set this on every application server in your environment by following the steps below:

1.On the administrative console, navigate to Servers > Server Types > WebSphere application servers
2.Click the server name, then navigate to Java and Process management > Process definition > Java Virtual Machine > Custom properties
3.Name: com.ibm.bpm.security.csrffilter.whitelist.
Value: *comma separated list of acceptable hostnames or domains*
4.Click OK, then Save the changes in the administrative console
5.Repeat the steps above for every server
6.Restart the servers to activate the changed value

For opening administrative console

http://ibpm85:9060/admin



Cell=ibpm85Cell01, Profile=DmgrProfile

Application servers

Messages

Changes have been made to your local configuration. You can:

- Save directly to the master configuration.
- Review changes before saving or discarding.

An option to synchronize the configuration across multiple nodes after saving can be enabled in Preferences.

The server may need to be restarted for these changes to take effect.

Application servers > AppClusterMember1 > Process definition > Java Virtual Machine > Custom properties > com.ibm.bpm.security.csrffilter.whitelist

Use this page to specify an arbitrary name and value pair. The value that is specified for the name and value pair is a string that can set internal system configuration properties.

Configuration

General Properties

* Name: com.ibm.bpm.security.csrffilter.white

* Value: ibpm85.eidikointernal.com

Description:

Apply | OK | Reset | Cancel

Alternatively, you can run the steps below in wsadmin to add the value to your environment. You will need to adapt the highlighted parts to your system:

```
 server = AdminConfig.getid("/Cell:PCCell1/Node:Node1/Server:server1/")
 jvm = AdminConfig.list('JavaVirtualMachine',server).split()[0]
 attr_name  = ['name', "com.ibm.bpm.security.csrffilter.whitelist.Referer"]
 attr_value = ['value', "hostname.com"] attr_required = ['required', "false"]
 attr_description = ['description', "whitelist of acceptable REFERER header values"]
 attr_list = [attr_name, attr_value, attr_required, attr_description]
 property=['systemProperties',[attr_list]]
 AdminConfig.modify(jvm, [property])
AdminConfig.save()
```

Example :
Goto path : /opt/IBM/WebSphere/AppServer/profiles/Node1Profile/bin
command: ./wsadmin.sh -host ibpm85 -port 8881 -lang jython -user bandaru -password sarasu10

```
 AdminTask.setBPMProperty(['-de', 'De1', '-name',
'ProcessServer.CsrfProtectionRefererWhitelist', '-value',
 'ibpm85.eidikointernal.com'])
```

2. ProcessServer.CsrfProtectionOriginWhitelist

* In a CSRF attack, the attacker prepares a malicious web site that causes the victim's browser to send a request to a vulnerable server.

* For simple requests (HTTP GET and POST with a Content-Type as used in plain HTML form submissions), the victim's browser will just send the request and include all cookies associated with the target URL.

* The vulnerable server has no means of determining whether the request was sent intentionally by the end user or the end user was tricked into sending that request.

* A widely effective protection is checking the HTTP request headers Referer and Origin against a whitelist. For cross-site requests, browsers include the URL of the current origin in the header.

* If you know that users access IBM BPM only by using bookmarks or links from some intranet portal, you can whitelist these by security hardening properties.

The Origin header is typically added if the request was sent as an XMLHttpRequest. Some security researchers have argued that it is more reliable than the Referer header, so it can also be restricted using a whitelist.

Command:
Goto path : /opt/IBM/WebSphere/AppServer/profiles/Node1Profile/bin
command: ./wsadmin.sh -host ibpm85 -port 8881 -lang jython -user bandaru -password sarasu10
AdminTask.setBPMProperty(['-de', 'De1', '-name', 'ProcessServer.CsrfProtectionOriginWhitelist', '-value', 'https://some.server.com**’,’https://some.other.server.com'])**

3. ProcessServer.XframeOptionsHeaderValue

When running a security scanning tool such as IBM Security AppScan it might show that WebSphere Application Server and Business Process Manager URLs such as
https://hostname:9043/ibm/console
https://hostname:4443/ProcessPortal
https://hostname:4443/ProcessAdmin
https://hostname:4443/bpc
shttps://hostname:4443/BusinessSpace
are vulnerable for Cross Frame Scripting-Click jacking - Cross Frame Scripting.

* Cross Frame Scripting-Click jacking - Cross Frame Scripting (XFS) is an attack that exploits the bug in specific browsers and captures the sensitive information from the legitimate users of the application.

* The attacker induces the browser for a user to navigate to a web page that the attacker controls, by loading a third-party page in an HTML frame and then the JavaScript executing in the attacker's page steals data from the third-party page.

* If you are using a webserver in front of your environment, you can add the X-Frame-Options header in IHS. To add it using IHS, you first need to ensure that the mod_headers module is loaded, as per below: LoadModule headers_module modules/mod_headers.so
Then, add the X-Frame-Options header using Header Set inside the virtualhost being used, as per below: Header set X-Frame-Options SAMEORIGIN
For example:

```
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
Listen 0.0.0.0:443
Header set X-Frame-Options SAMEORIGIN
```

* If you are not using IHS, then the application itself should set the header. You can add the X-Frame-Options to HTTP response header using deployment descriptor (web.xml). You may need to engage your application developers on how to configure it in the application.

Command :
Goto path : /opt/IBM/WebSphere/AppServer/profiles/Node1Profile/bin
command: ./wsadmin.sh -host ibpm85 -port 8881 -lang jython -user bandaru -password sarasu10
Then
AdminTask.setBPMProperty(['-de', 'De1', '-name', 'ProcessServer.XFrameOptionsHeaderValue', '-value', 'SAMEORIGIN'])

* This setting avoids manipulating web.xml as it will be read by BPM application code to cause a ServletFilter in front of protected resources to add this header. The IHS solution is complete, though. It will inject the header in any traffic.

*This includes SOAP over HTTP traffic where it is not necessary, but it also catches some unprotected resources that are not behind BPM's ServletFilter.

You set the value of the X-Frame-Options header field of the HTTP response with this property.IBM BPMreturns this value to client requests. The value disallows browsers to embed IBM BPM user interfaces in iframes and, thus, mitigates potential clickjacking attacks. You can set the following values:
•**DENY:** This value specifies that IBM BPM must not be embedded in iframes.
Note: This value does not work with client-side human services and IBM Process Portal.
•**SAMEORIGIN:** This value is recommended for most scenarios.IBM BPM user interfaces can be embedded in iframes, if the iframe element is served from the same origin as the IBM BPM user interface.
•**ALLOW-FROM** https://example.com/: This value allows the browser to embed IBM BPM in iframes served from the https://example.com/ URI.

4. <u>ProcessServer.ContentSecurityPolicyHeaderValue</u>

* IBM Business Process Manager (BPM) does not enable you to set the X-FRAME-OPTIONS header that allows administrators to define whether and how IBM BPM user interface content can be embedded in HTML iframes on either the same or other domains.

* You set the value of the Content-Security-Policy and X-Content-Security-Policy HTTP response header fields with this property.
* IBM BPM returns this value to client requests. The value instructs the browser to load and run assets in the context of IBM BPM user interfaces only from a set of whitelisted origins.

* For example, the policy default-src 'unsafe-inline' 'unsafe-eval' https://bpm1.ibm.com https://bpm1; img-src data: https://bpm1.ibm.com https://bpm1instructs the browser to retrieve images from data: URIs and from URIs that meet one of the two whitelisted URI patterns.

<u>Synatx:</u>

Goto path : /opt/IBM/WebSphere/AppServer/profiles/Node1Profile/bin
command: ./wsadmin.sh -host ibpm85 -port 8881 -lang jython -user bandaru -password sarasu10
Then

```
AdminTask.setBPMProperty(['-de', 'De1', '-name',
'ProcessServer.ContentSecurityPolicyHeaderValue', '-value', "default-src 'self'
'unsafe-inline' 'unsafe-eval' https://fmtc4090.boeblingen.de.ibm.com"])
```

4. ProcessServer.StrictTransportSecurityHeaderValue

HTTP Strict Transport Security (HSTS)
* This specification defines a mechanism enabling web sites to declare themselves accessible only via secure connections and/or for users to be able to direct their user agent(s) to interact with given sites only over secure connections. This overall policy is referred to as HTTP Strict Transport Security (HSTS).
* The policy is declared by websites via the Strict-Transport-Security HTTP response header field and/or by other means, such as user agent configuration.

You set the value of the Strict-Transport-Security HTTP response header field with this property. IBM BPM returns this value to client requests. The value instructs the browser to upgrade any http:// link to the server to an https:// link.

Syntax:

Goto path : /opt/IBM/WebSphere/AppServer/profiles/Node1Profile/bin

command: ./wsadmin.sh -host ibpm85 -port 8881 -lang jython -user bandaru -password sarasu10

Then

```
AdminTask.setBPMProperty(['-de', 'De1', '-name',
   'ProcessServer.StrictTransportSecurityHeaderValue', '-value', "max-age=100;
   includeSubDomains"])
```

5. ProcessServer.XxssProtectionHeaderValue

***** The HTTP **X-XSS-Protection** response header is a feature of Internet Explorer, Chrome and Safari that stops pages from loading when they detect reflected cross-site scripting (XSS) attacks.

***** Cross-site scripting (XSS) is a computer security vulnerability that allows malicious attackers to inject client-side script into web pages viewed by other users.

* You can use the **Cross-site Scripting Filter** setting to check all HTTP GET requests sent to IBM® OpenPages® GRC Platform.

* The **Cross-site Scripting Filter** setting enables basic filtering of common attacks. The **Advanced XSS Filter** setting turns on more aggressive filtering of JavaScript actions. The **IE XSS Filter** setting is used to set the X-XSS-Protection header on a request. However, the preferred approach is to use the **X-XSS-Protection** header setting

*****You set the value of the X-XSS-Protection HTTP response header field with this property. IBM BPMreturns this value to client requests.
* The value instructs the browser to enable its built-in cross-site scripting protection, independent of the user's configuration.
* Setting this property can be useful if the Internet Explorer browser categorizes IBM BPM to run in the intranet zone, which applies weaker browser security settings.

Syntax:

Goto path : /opt/IBM/WebSphere/AppServer/profiles/Node1Profile/bin
command: ./wsadmin.sh -host ibpm85 -port 8881 -lang jython -user bandaru -password sarasu10
Then

```
AdminTask.setBPMProperty(['-de', 'De1', '-name',
   'ProcessServer.XXssProtectionHeaderValue', '-value', "1;
   mode=block"])
```

a sample value is 1 and mode = block.

6. ProcessServer.XcontentTypeOptionsHeaderValue

* The X-Content-Type-Options response HTTP header is a marker used by the server to indicate that the MIME types advertised in the Content-Type headers should not be changed and be followed. This allows to opt-out of MIME type sniffing, or, in other words, it is a way to say that the webmasters knew what they were doing.

* You set the value of the X-Content-Type-Options HTTP response header field with this property. IBM BPMreturns this value to client requests.
* For example, a value of nosniff instructs browsers to disable MIME-type sniffing (a technique that tries to determine a suitable rendering strategy for server content based on the contents of the HTTP response).
* MIME-type sniffing can interfere with the explicit overwriting of MIME types in IBM BPM document downloads and, thus, re-run JavaScript even if IBM BPM explicitly set the Content-Type header to force the download window.

Syntax:

Goto path : /opt/IBM/WebSphere/AppServer/profiles/Node1Profile/bin
command: ./wsadmin.sh -host ibpm85 -port 8881 -lang jython -user bandaru -password sarasu10
Then

```
AdminTask.setBPMProperty(['-de', 'De1', '-name',
   'ProcessServer.XContentTypeOptionsHeaderValue', '-value', "nosniff"])
```

7. ProcessServer.CsrfSessionTokenSaltProcessServer.CsrfSessionTokenProtectedUris

* Cross Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions. For example, an attacker prepares a website that causes the victim's browser to send a request to a vulnerable server, such as https://sample.bank.com/transfermoney?toAccount=12345&amount=300.

* Browser usually include an HTTP request header REFERER when sending GET or POST requests to a server based on a link or form on a website of a different server.

* All web applications in IBM Business Process Manager (BPM) support protection against Cross-Site Request Forgery (CSRF) by using a REFERER header whitelist

You enable session-specific tokens to mitigate CSRF with these properties.

•To generate identical CSRF protection tokens on all cluster members without sharing the generated tokens, use the ***ProcessServer.CsrfSessionTokenSalt*** property. To activate the CSRF protection token, set any value for *ProcessServer.CsrfSessionTokenSalt*.

•For the server to check for CSRF protection, use the ***ProcessServer.CsrfSessionTokenProtectedUris*****property**. The only supported value for this property is /teamworks/ajaxCoach, which enforces server-side checking of CSRF protection tokens for this exact URL.

Enable the CSRF security property in wsadmin by adding this code block:

Syntax:

Goto path : /opt/IBM/WebSphere/AppServer/profiles/Node1Profile/bin

command:

```
./wsadmin.sh -host ibpm85 -port 8881 -lang jython -user bandaru -password sarasu10
```

Then

```
AdminTask.setBPMProperty(['-de', 'De1', '-name', 'ProcessServer.CsrfSessionTokenSalt', '-
value', 'verySecret'])
AdminTask.setBPMProperty(['-de', 'De1', '-name',
'ProcessServer.CsrfSessionTokenProtectedUris', '-value', '/teamworks/ajaxCoach,/teamworks/
/ajaxCoach'])
AdminConfig.save()
```