

# *Bravura Privilege*

## Web App Privileged Sign-on Disclosure

This document shows you how to configure and use the web app privileged sign-on disclosure.

- Objective
- Solution
- Deployment
- Use case: Configuring a website disclosure configuration
- Use case: Configuring a team account to use web app privileged sign-on
- Use case: Disclosing website access to a user
- Configuration file
- Utility: pswxdom2webapp

### Terminology

**Web application administrator** User who has access to website disclosure configuration pre-defined request (PDR)s. This person will need to be a member of the PAM\_TEAM\_ADMINS user class, and will need to be configured by a product administrator.

**Hitachi ID browser extension** An extension that allows the web app privileged sign-on disclosure to broker access to the configured website.

## 1 Objective

Organizations need to broker access to hundreds and thousands of websites. Users need access from any browser without having to provide administrator credentials.

## 2 Solution

The web app privileged sign-on disclosure allows organizations to broker access to hundreds and thousands of web applications. The access settings for web applications are configured in a json file that is loaded when the web application is created in the product. This disclosure launches a new web page tab within the same browser and provides automatic login to the website without need to enter administrator credentials for the managed account using the configuration file.

## 3 Requirements

In order to use the web app privileged sign-on disclosure, a Google Chrome or Microsoft Edge (Chromium) browser with Hitachi ID Browser Extension 1.1.4 or newer needs to be installed. This extension can be obtained from the Chrome Web Store.

## 4 Deployment

The following components must be installed in addition to the *Hitachi ID Bravura Pattern: Privileged Access Edition* configuration:

- Scenario.pam\_webapp\_management
- Scenario.pam\_webapp\_social (optional)
- Configuration file (json) - See [Configuration file](#)

## 5 Use case: Configuring a website disclosure configuration

Web application administrators (web app admins) can create, update and delete website disclosure configurations for broking access to specific websites.

### Additional requirements

This use case assumes that:

- A web app admin has been added to the PAM\_TEAM\_ADMINS user class and thus has the privilege to configure website disclosure configurations.
- A configuration file is available and complete.

### Create a website disclosure configuration

To create a website disclosure configuration:

1. Log in to *Bravura Privilege* as a web app admin.
2. Select **Manage Resources**.
3. Select **Website Disclosure Configuration: Create**.
4. Enter a **Name**, **Description**, and select a **Configuration file**.

The screenshot shows a web interface for creating a website disclosure configuration. The main form area is titled 'Website disclosure configuration' and contains three labeled input fields: 'Name \*' with the value 'webapp-sample', 'Description \*' with the value 'Sample web application', and 'Configuration file (json) \*' with the value 'webapps.json'. To the right of the form is a 'Details' sidebar that lists the configuration attributes: 'Name', 'Description', and 'Configuration file (json)'. At the bottom right of the interface is a 'Submit' button.

5. Click **Submit**.

The request should be automatically approved.

## Update a website disclosure configuration

To update a website disclosure configuration:

1. Log in to *Bravura Privilege* as a web app admin.
2. Select **Manage Resources**.
3. Select **Website Disclosure Configuration: Update**.
4. Select the website disclosure configuration you want to update.
5. Click **Next**.
6. Update the **Name**, **Description**, and/or **Configuration file**.

**Note:** Each website disclosure configuration is uniquely identified by Hitachi Bravura Privilege, so changing the name will not modify which website disclosure configuration is used.

Website Disclosure Configuration: Update

Website disclosure configuration

Name \*  
webapp-sample

Description \*  
Sample web application

Configuration file (json)

Details

Select a website disclosure configuration  
1 attribute changed  
+ Select a website disclosure configuration

Website disclosure configuration  
2 attributes changed  
+ Name  
+ Description

Requester notes:

Previous Submit

7. Click **Submit**.

The request should be automatically approved.

## Delete a website disclosure configuration

To delete a website disclosure configuration:

1. Log in to *Bravura Privilege* as a web app admin.
2. Select **Manage Resources**.
3. Select **Website Disclosure Configuration: Delete**.
4. Select the website disclosure configuration you want to delete.

Website Disclosure Configuration: Delete

Select a website disclosure configuration

Select a website disclosure configuration \*

webapp-sample - Sample web application X

Details

Select a website disclosure configuration

1 attribute changed

+ Select a website disclosure configuration

Confirm deletion

Requester notes:

Next Submit

5. Click **Next**.
6. Confirm delete.

**Note:** Number of accounts using this website disclosure configuration will be displayed.

Website Disclosure Configuration: Delete

Confirm deletion

0 account(s) will be affected.

Are you sure that you want to delete everything associated with the website disclosure configuration?

☐

Details

Select a website disclosure configuration

1 attribute changed

+ Select a website disclosure configuration

Confirm deletion

1 attribute changed

Are you sure that you want to delete everything associated with the website disclosure configuration?

Requester notes:

Previous Submit

7. Click **Submit**.

The request should be automatically approved.

## 6 Use case: Configuring a team account to use web app privileged sign-on

Once website disclosure configurations are configured, team accounts can select them as a disclosure option. By default, this is only available for team vault accounts.

### Additional requirements

This use case assumes that:

- A team has been created and configured
- A vault trustee has been configured for a team
- A team vault has been configured
- A website disclosure configuration is configured and available

**Note:** Built-in social website disclosure configurations for GMail, Facebook, and Twitter are available. Install the Scenario.pam\_webapp\_social component to use these.

### Create a vault account to use web app privileged sign-on

To create a team account:

1. Log in to *Bravura Privilege* as a vault trustee.
2. Select **Manage Resources**.
3. Select **Vault Account: Create**.
4. Select a managed system.
5. Click **Next**.
6. Enter **Account Name**.
7. Enter **Account Password** and confirm.
8. Select one or more **Available website disclosure configurations**.
9. Select **Web app privileged sign-on** from the **Available website disclosure methods**.

**Vault Account: Create**

System ID \*  
TVAULT

System Name \*  
Team Database Vault

System Team \*  
Vault-Team

Account Name \*  
testacct

Account Password \*  
.....  
.....

Available website disclosure configurations  
webapp-sample - Sample web application X

Available website disclosure methods  
Web app privileged sign-on X

**Details**

Select Resource  
1 attribute changed  
+ Select a Managed System

Account Information  
8 attributes changed  
+ System ID  
+ System Name  
+ System Team  
~ Show more

Requester notes:  
.....

Previous Submit

10. Click **Submit**.

The request should be automatically approved if submitted by the team's vault trustee. Otherwise the appropriate trustee will need to approve the request.

### Update a vault account to use different website disclosure configurations

To update a team account:

1. Log in to *Bravura Privilege* as a vault trustee.
2. Select **Manage Resources**.
3. Select **Vault Account: Update**.
4. Select a managed account.
5. Click **Next**.
6. Click **Next**.
7. Select/deselect one or more **Available website disclosure configurations**.

**Vault Account: Update**

Disclosure Attributes

Available website disclosure configurations  
webapp-sample - Sample web application X

Available website disclosure methods  
Web app privileged sign-on X

Copy Password  
☒

View Password  
☒

**Details**

1 attribute changed  
+ Select a Managed Account

Account Attributes for testacct  
5 attributes changed  
+ System ID  
+ System Name  
+ System Team  
~ Show more

Disclosure Attributes  
6 attributes changed  
+ Available website disclosure configurations  
+ Available website disclosure methods  
+ Single Sign-On connection  
~ Show more

Previous Submit

8. Click **Submit**.

The request should be automatically approved if submitted by the team's vault trustee. Otherwise the appropriate trustee will need to approve the request.

## 7 Use case: Disclosing website access to a user

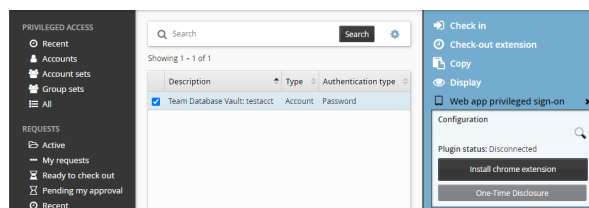
### Additional requirements


This use case assumes that:

- A team account has been configured to broker web app privileged sign-on.
- A user is a member of the team with requester privileges.
- The user logs into *Bravura Privilege* using SSL (HTTPS) from a Google Chrome or Microsoft Edge (Chromium) browser. See [SSL enforcement](#) for more details.

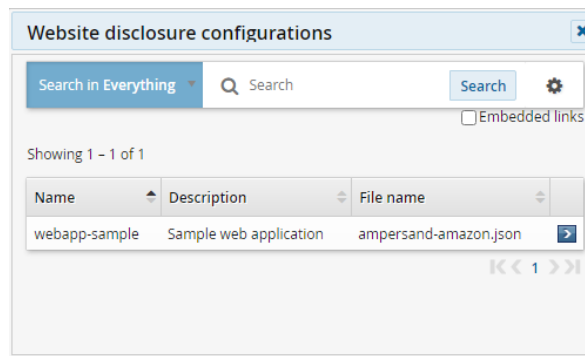
To request, check out and disclose access:

1. Log in to *Bravura Privilege* as an end user.
2. Select **Privileged access**.
3. Search and select the account with access.
4. Submit a request for access. Wait for approval as required through your organization's process.
5. Check out the account access.
6. Expand **Web app privileged sign-on**.

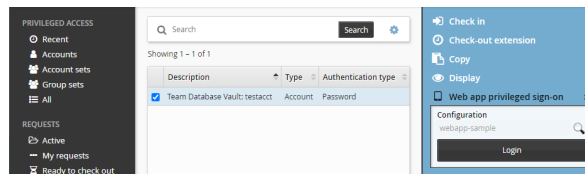


7. Install the Hitachi ID Browser Extension from the Chrome Web Store.
8. Click the search icon .
9. Select **Configuration file**.





10. Click **Login**.



**Note:** Pop-up blockers may prevent this feature from launching a new tab.

## 8 SSL enforcement

By default, users can only disclose website disclosure configurations if accessing *Bravura Privilege* using SSL (HTTPS). If the insecure HTTP method is used, the user will see a 'Web App disclosure enforces SSL' message and will not be able to access the website disclosure configuration.

To allow disclosing of website disclosure configurations regardless of protocol (not recommended):

1. Click **Manage the system** → **Maintenance** → **System variables**.
2. Set **PAM WEBAPP ENFORCE SSL** to **Disabled**.
3. Click **Update**.

# Appendices

## A Configuration file

The configuration file for the web app privileged sign-on disclosure requires a JSON file with the following structure:

```
{
  "info": {
    "url": "https://webpage/"
  },
  "username": {
    "order": 1,
    "type": "CSS",
    "value": "input[name='session[username_or_email]']",
    "input_value": "%username%",
    "keypress_event": true,
    "click": false,
    "settle_time_before": 5,
    "settle_time_after": 0,
    "stop_on_fail": false
  },
  "password": {
    "order": 2,
    "type": "CSS",
    "value": "input[name='session[password]']",
    "input_value": "%password%" ,
    "keypress_event": true,
    "click": false,
    "settle_time_before": 0,
    "settle_time_after": 0,
    "stop_on_fail": false
  },
  "submit": {
    "order": 3,
    "type": "CSS",
    "value": "[role='button'][type='submit']",
    "input_value": "" ,
    "keypress_event": false,
    "click": true,
    "settle_time_before": 0,
    "settle_time_after": 0,
    "stop_on_fail": false
  }
}
```

Where:

**"Info"**["url"] is the url the web app will open the webpage on.

**"username"** , **"password"**, and **"submit"** action keys with a structure which determines how the browser will interact with the webpage.

You can have as many action keys as needed to interact with the webpage tab

**"order"** order number the browser extension will act on.

**"type"** the search method the browser extension will use to find an element that matches "value" on the webpage. The supported types are:

- "XPATH": find an element using XPATH that matches the "value"
- "CSS": find an element using CSS that matches the "value"

- "ID": find an element with "id" that matches the "value"
- "NAME": find the first element with "name" that matches the "value"
- "CLASS": find the first element with the class that matches the "value"

"value" browser extension will use to look the webpage by "type".

"input\_value" the value we will look for in the search to identify the input field.

"keypress\_event" a flag that is required, but not yet used.

"click" a flag that determines if the browser extension will click on the "input\_value" element.

"settle\_time\_before" the time in seconds the browser extension waits before accessing this element.

"settle\_time\_after" the time in seconds the browser extension waits after accessing this element.

"stop\_on\_fail" a flag that stops the continued execution if the browser extension fails to find the element set by "value".

## A.1 Sample configuration file for web app privileged sign-on (twitter.json)

```
{
  "info": {
    "url": "https://twitter.com/"
  },
  "username": {
    "order": 1,
    "type": "CSS",
    "value": "input[name='session[username_or_email]']",
    "input_value": "%username%",
    "keypress_event": true,
    "click": false,
    "settle_time_before": 5,
    "settle_time_after": 0,
    "stop_on_fail": false
  },
  "password": {
    "order": 2,
    "type": "CSS",
    "value": "input[name='session[password]']",
    "input_value": "%password%" ,
    "keypress_event": true,
    "click": false,
    "settle_time_before": 0,
    "settle_time_after": 0,
    "stop_on_fail": false
  },
  "submit": {
    "order": 3,
    "type": "CSS",
    "value": "[role='button'][data-testid='LoginForm_Login_Button']",
    "input_value": "" ,
    "keypress_event": false,
    "click": true,
  }
}
```

```

    "settle_time_before": 0,
    "settle_time_after": 0,
    "stop_on_fail": false
  }
}

```

## B Utility: pswxdom2webapp

The **pswxdom2webapp** utility converts existing Browser Driver (pswxdom) configuration to JSON that can be used for web apps. The utility locates all Browser Driver access disclosure plugins, including ones created globally, as well as those overridden in the managed system policy (MSP) level. By default, a corresponding web app will be generated for each converted Browser Driver access disclosure. Browser Driver access disclosures used in the MSP-level will not be converted if none of the disclosure attributes values are overridden.

The **pswxdom2webapp** utility can be found in the util directory.

The following Browser Driver disclosure attribute settings are converted:

- checkboxdata
- constfielddata
- formatted username
- passwordfieldids
- settletime
- submitbuttonids
- url
- usernamefieldids

The following Browser Driver disclosure attribute settings are *not* converted:

- denypopups
- formatted title
- height
- uicontrols
- width
- simulatekeypress\*

Notes:

- simulatekeypress will be supported, but not in Phase 1. A warning message will be given when pswxdom2webapp finds a Browser Driver disclosure that has this setting enabled. A web app will be created, but running it may cause issues if the website relies on keypresses to log in.
- If a semicolon is used as a submitbuttonids rather than an HTML tag, it will no longer work on web app. This will need to be replaced with an explicit HTML tag.

### B.1 Usage

```
pswxdom2webapp.exe [-prefix] [-file] [-force]
```

Table 1: pswxdom2webapp arguments

Argument	Description
-prefix	Prefix the web app's name with the specified value.
-file	Convert Browser Driver disclosure configuration to JSON files. This is added to <i>&lt;instance&gt;/webappfiles</i> . This does not commit changes to the database. The JSON files generated can be used to manually create web apps.
-force	Override existing changes made to the database or converted web app files.

Converted global-level Browser Driver disclosure plug-ins will be named: *<disclosure plugin>*. If the -prefix argument is used, they will be named: *<prefix><disclosure-plugin>*. Description of the web app will be: Webappjson generated for *<disclosure plugin>*.

Converted MSP-level Browser Driver disclosure plugins will be named: *<MSP>-<disclosure plugin>*. If the -prefix argument is used, they will be named: *<prefix><MSP>-<disclosure-plugin>*. Description of the web app will be: Webappjson generated for *<MSP> / <disclosure plugin>*.

If a web app was already generated for a specific Browser Driver disclosure using pswxdom2webapp, it will not be generated again unless -force is used.

## B.2 Examples

- To convert all browser driver disclosures to web app and adding it to the database, assuming none of the browser driver disclosures are converted previously:

```
pswxdom2webapp.exe
```

- To convert all browser driver disclosures to web app and add them to the database, regardless if they have been already converted using pswxdom2webapp :

```
pswxdom2webapp.exe -force
```

- To convert all browser driver disclosures to web app, but only create JSON configuration files instead of creating web app entries in the database:

```
pswxdom2webapp.exe -file
```

- To convert all browser driver disclosures to web app, but only create JSON configuration files regardless if they have been already converted using pswxdom2webapp:

```
pswxdom2webapp.exe -file -force
```