Response to Fortify Scan of TAS Application dated May 11, 2018

Developer: Patrick Whalen   
email: patrick.whalen@va.gov

Note: TAS Application has changed from using the “min.js” file to the un-minified file (tableau-2.2.2.js) for easier reading in the future.

# Issues

## tableau-2.2.2.js, line 7015 (Key Management: Hardcoded Encryption Key) (13 Issues on line 7015)

7015 if (entry.key !== 'embed' && entry.key !== 'height' && entry.key !== 'width' &&

entry.key !== 'device' && entry.key !== 'autoSize' && entry.key !== 'hideTabs' &&

entry.key !== 'hideToolbar' && entry.key !== 'onFirstInteractive' && entry.key !==

'onFirstVizSizeKnown' && entry.key !== 'toolbarPosition' && entry.key !==

'instanceIdToClone' && entry.key !== 'navType' && entry.key !==

'display\_static\_image') {

Response:

This is a false positive. The “key” is merely the key in a JavaScript object’s key-value pair. No security-context encryption key is hardcoded in the code.

## tableau-2.2.2.js, line 4572 (HTML5: Overly Permissive Message Posting Policy)

4572 iframe.contentWindow.postMessage(message, '\*');

Response:

This is a false positive. “postMessage” function is used for controlling iFrame and returns simple iFrame status information. The calling code limits these response messages so no spoofing, data theft, relay and other attacks are possible.

## tableau-2.2.2.js, line 8626 (HTML5: Overly Permissive Message Posting Policy)

8626 this.$iframe.contentWindow.postMessage(command.serialize(), '\*');

Response:

This is a false positive. “postMessage” function is used for controlling iFrame and returns simple iFrame status information. The calling code limits these response messages so no spoofing, data theft, relay and other attacks are possible.

## tableau-2.2.2.js, line 8612 (HTML5: Overly Permissive Message Posting Policy)

8612 this.$iframe.contentWindow.postMessage(command.serialize(), '\*');

Response:

This is a false positive. “postMessage” function is used for controlling iFrame and returns simple iFrame status information. The calling code limits these response messages so no spoofing, data theft, relay and other attacks are possible.

## tableau-2.2.2.js, line 4608 (HTML5: Overly Permissive Message Posting Policy)

4608 messageEvent.source.postMessage(bootstrapCommand.serialize(), '\*');

Response:

This is a false positive. “postMessage” function is used for controlling iFrame and returns simple iFrame status information. The calling code limits these response messages so no spoofing, data theft, relay and other attacks are possible.

## tableau-2.2.2.js, line 4576 (HTML5: Overly Permissive Message Posting Policy)

4576 iframe.contentWindow.postMessage(message, '\*');

Response:

This is a false positive. “postMessage” function is used for controlling iFrame and returns simple iFrame status information. The calling code limits these response messages so no spoofing, data theft, relay and other attacks are possible.

## main-content.component.html, line 10 (Cross-Site Request Forgery)

.<.form class="usa-form-small"

Response:

This is a false positive. TAS Application forms do not use postbacks (non-web 2.0/no-AJAX) to communicate with server. TAS Application does not manage its own session cookies. Forms are used for page structure.

## main-content.component.html, line 14 (Cross-Site Request Forgery)

<form class="usa-form-small" \*ngIf="reportsConfig && !isSaving">

Response:

This is a false positive. TAS Application forms do not use postbacks (non-web 2.0/no-AJAX) to communicate with server. TAS Application does not manage its own session cookies. Forms are used for page structure.

## tableau-2.2.2.js, line 116 (Insecure Randomness)

116 return obj.$\_\_hashCode\_\_ || (obj.$\_\_hashCode\_\_ = (Math.random() \* 0x100000000) | 0);

Response:

This is a false positive. Function is not used in a security-sensitive context. This function uses a random number to build a hash table for objects inside Tableau report.

## uswds.js, line 1771 (Password Management: Password in Comment)

1771 /\*\*

1772 \* Component that decorates an HTML element with the ability to toggle the

1773 \* masked state of an input field (like a password) when clicked.

Response:

This is a false positive. The word “password” is in a comment. No password variable is set in the code or included in a comment.