opentext™

Fortify Security Report STIG 5.3

3/5/24

Jason Pyeron

Executive Summary

Issues Overview

On Mar 5, 2024, a source code review was performed over the modheader code base. 14 files, 537 LOC (Executable) were scanned and reviewed for defects that could lead to potential security vulnerabilities. A total of 3 reviewed findings were uncovered during the analysis.

Issues by STIG 5.3

APSC-DV-002560 CAT I	1
APSC-DV-002490 CAT I, APSC-DV-002560 CAT I	1
APSC-DV-002010 CAT II, APSC-DV-002050 CAT II	1

Recommendations and Conclusions

The Issues Category section provides Fortify recommendations for addressing issues at a generic level. The recommendations for specific fixes can be extrapolated from those generic recommendations by the development group.

Project Summary

Code Base Summary

Code location: C:/projects/modheader
Number of Files: 14
Lines of Code: 537
Build Label: <No Build Label>

Scan Information

Scan time: 01:17
SCA Engine version: 23.1.0.0136
Machine Name: blackfat
Username running scan: jpyeron

Results Certification

Results Certification Valid

Details:

Results Signature:

SCA Analysis Results has Valid signature

Rules Signature:

There were no custom rules used in this scan

Attack Surface

Attack Surface:

Filter Set Summary

Current Enabled Filter Set:
Security Auditor View

Filter Set Details:

Folder Filters:
If [fortify priority order] contains critical Then set folder to Critical
If [fortify priority order] contains high Then set folder to High
If [fortify priority order] contains medium Then set folder to Medium
If [fortify priority order] contains low Then set folder to Low

Audit Guide Summary

Audit guide not enabled

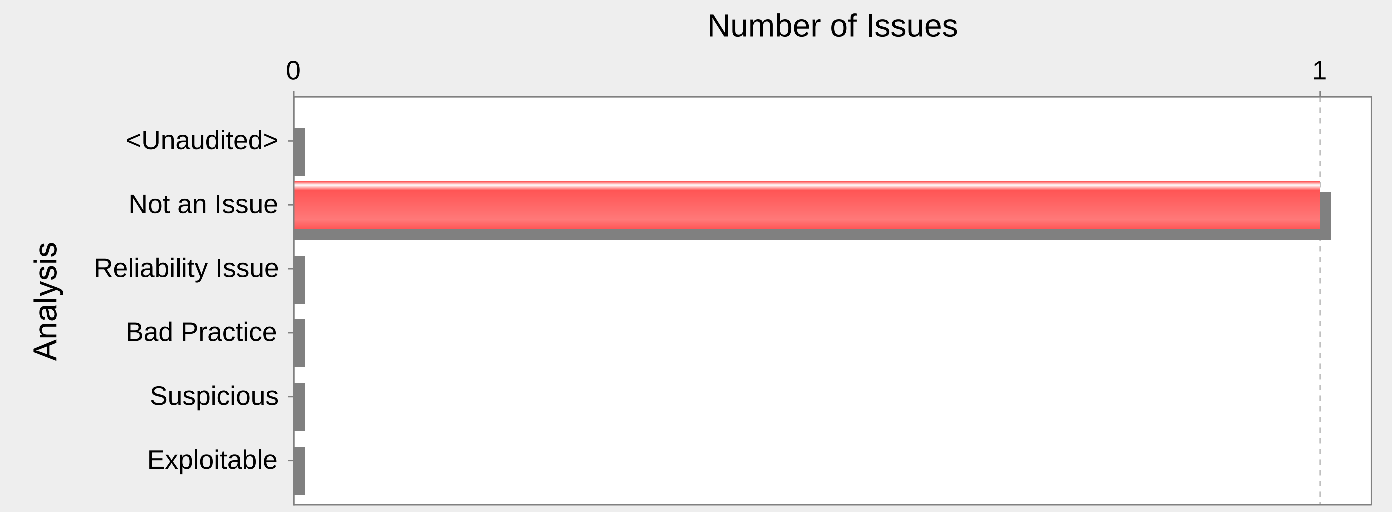
Results Outline

Overall number of results

The scan found 3 issues.

Vulnerability Examples by Category

Category: Cross-Site Scripting: DOM (1 Issues)



Abstract:

The method addFilter() in main.js sends unvalidated data to a web browser on line 48, which can result in the browser executing malicious code.

Explanation:

Cross-site scripting (XSS) vulnerabilities occur when:

1. Data enters a web application through an untrusted source. In the case of DOM-based XSS, data is read from a URL parameter or other value within the browser and written back into the page with client-side code. In the case of reflected XSS, the untrusted source is typically a web request, while in the case of persisted (also known as stored) XSS it is typically a database or other back-end data store.

2. The data is included in dynamic content that is sent to a web user without validation. In the case of DOM-based XSS, malicious content is executed as part of DOM (Document Object Model) creation, whenever the victim's browser parses the HTML page.

The malicious content sent to the web browser often takes the form of a JavaScript segment, but can also include HTML, Flash or any other type of code that the browser executes. The variety of attacks based on XSS is almost limitless, but they commonly include transmitting private data such as cookies or other session information to the attacker, redirecting the victim to web content controlled by the attacker, or performing other malicious operations on the user's machine under the guise of the vulnerable site.

Example 1: The following JavaScript code segment reads an employee ID, eid, from a URL and displays it to the user.

```
<SCRIPT>
var pos=document.URL.indexOf("eid=")+4;
document.write(document.URL.substring(pos,document.URL.length));
</SCRIPT>
```

Example 2: Consider the HTML form:

```
<div id="myDiv">
Employee ID: <input type="text" id="eid"><br>
...
<button>Show results</button>
</div>
<div id="resultsDiv">
...
</div>
```

The following jQuery code segment reads an employee ID from the form, and displays it to the user.

```
$(document).ready(function(){
$("#myDiv").on("click", "button", function(){
var eid = $("#eid").val();
$("resultsDiv").append(eid);
...
});
});
```

These code examples operate correctly if the employee ID from the text input with ID eid contains only standard alphanumeric text. If eid has a value that includes metacharacters or source code, then the code will be executed by the web browser as it displays the HTTP response.

Example 3: The following code shows an example of a DOM-based XSS within a React application:

```
let element = JSON.parse(getUntrustedInput());
ReactDOM.render(<App>
{element}
</App>);
```

In Example 3, if an attacker can control the entire JSON object retrieved from getUntrustedInput(), they may be able to make React render element as a component, and therefore can pass an object with dangerouslySetInnerHTML with their own controlled value, a typical cross-site scripting attack.

Initially these might not appear to be much of a vulnerability. After all, why would someone provide input containing malicious code to run on their own computer? The real danger is that an attacker will create the malicious URL, then use email or social engineering tricks to lure victims into visiting a link to the URL. When victims click the link, they unwittingly reflect the malicious content through the vulnerable web application back to their own computers. This mechanism of exploiting vulnerable web applications is known as Reflected XSS.

As the example demonstrates, XSS vulnerabilities are caused by code that includes unvalidated data in an HTTP response. There are three vectors by which an XSS attack can reach a victim:

- Data is read directly from the HTTP request and reflected back in the HTTP response. Reflected XSS exploits occur when an attacker causes a user to supply dangerous content to a vulnerable web application, which is then reflected back to the user and executed by the web browser. The most common mechanism for delivering malicious content is to include it as a parameter in a URL that is posted publicly or emailed directly to victims. URLs constructed in this manner constitute the core of many phishing schemes, whereby an attacker convinces victims to visit a URL that refers to a vulnerable site. After the site reflects the attacker's content back to the user, the content is executed and proceeds to transfer private information, such as cookies that might include session information, from the user's machine to the attacker or perform other nefarious activities.

- The application stores dangerous data in a database or other trusted data store. The dangerous data is subsequently read back into the application and included in dynamic content. Persistent XSS exploits occur when an attacker injects dangerous content into a data store that is later read and included in dynamic content. From an attacker's perspective, the optimal place to inject malicious content is in an area that is displayed to either many users or particularly interesting users. Interesting users typically have elevated privileges in the application or interact with sensitive data that is valuable to the attacker. If one of these users executes malicious content, the attacker may be able to perform privileged operations on behalf of the user or gain access to sensitive data belonging to the user.

- A source outside the application stores dangerous data in a database or other data store, and the dangerous data is subsequently read back into the application as trusted data and included in dynamic content.

Recommendations:

The solution to prevent XSS is to ensure that validation occurs in the required places and that relevant properties are set to prevent vulnerabilities.

Because XSS vulnerabilities occur when an application includes malicious data in its output, one logical approach is to validate data immediately before it leaves the application. However, because web applications often have complex and intricate code for generating dynamic content, this method is prone to errors of omission (missing validation). An effective way to mitigate this risk is to also perform input validation for XSS.

Web applications must validate all input to prevent other vulnerabilities, such as SQL injection, so augmenting an application's existing input validation mechanism to include checks for XSS is generally relatively easy. Despite its value, input validation for XSS does not take the place of rigorous output validation. An application might accept input through a shared data store or other trusted source, and that data store might accept input from a source that does not perform adequate input validation. Therefore, the application cannot implicitly rely on the safety of this or any other data. This means that the best way to prevent XSS vulnerabilities is to validate everything that enters the application and leaves the application destined for the user.

The most secure approach to validation for XSS is to create an allow list of safe characters that can appear in HTTP content and accept input composed exclusively of characters in the approved set. For example, a valid username might only include alphanumeric characters or a phone number might only include digits 0-9. However, this solution is often infeasible in web applications because many characters that have special meaning to the browser must be considered valid input after they are encoded, such as a web design bulletin board that must accept HTML fragments from its users.

A more flexible, but less secure approach is to implement a deny list, which selectively rejects or escapes potentially dangerous characters before using the input. To form such a list, you first need to understand the set of characters that hold special meaning for web browsers. Although the HTML standard defines which characters have special meaning, many web browsers try to correct common mistakes in HTML and might treat other characters as special in certain contexts. This is why we do not recommend the use of deny lists as a means to prevent XSS. The CERT(R) Coordination Center at the Software Engineering Institute at Carnegie Mellon University provides the following details about special characters in various contexts [1]:

In the content of a block-level element (in the middle of a paragraph of text):

- "<" is special because it introduces a tag.

- "&" is special because it introduces a character entity.

- ">" is special because some browsers treat it as special, on the assumption that the author of the page intended to include an opening "<", but omitted it in error.

The following principles apply to attribute values:

- In attribute values enclosed in double quotes, the double quotes are special because they mark the end of the attribute value.

- In attribute values enclosed in single quote, the single quotes are special because they mark the end of the attribute value.

- In attribute values without any quotes, white-space characters, such as space and tab, are special.

- "&" is special when used with certain attributes, because it introduces a character entity.

In URLs, for example, a search engine might provide a link within the results page that the user can click to re-run the search. This can be implemented by encoding the search query inside the URL, which introduces additional special characters:

- Space, tab, and new line are special because they mark the end of the URL.

- "&" is special because it either introduces a character entity or separates CGI parameters.

- Non-ASCII characters (that is, everything greater than 127 in the ISO-8859-1 encoding) are not allowed in URLs, so they are considered to be special in this context.

- The "%" symbol must be filtered from input anywhere parameters encoded with HTTP escape sequences are decoded by server-side code. For example, "%" must be filtered if input such as "%68%65%6C%6C%6F" becomes "hello" when it appears on the web page.

Within the body of a <SCRIPT> </SCRIPT>:

- Semicolons, parentheses, curly braces, and new line characters must be filtered out in situations where text could be inserted directly into a pre-existing script tag.

Server-side scripts:

- Server-side scripts that convert any exclamation characters (!) in input to double-quote characters (") on output might require additional filtering.

Other possibilities:

- If an attacker submits a request in UTF-7, the special character '<' appears as '+ADw-' and might bypass filtering. If the output is included in a page that does not explicitly specify an encoding format, then some browsers try to intelligently identify the encoding based on the content (in this case, UTF-7).

After you identify the correct points in an application to perform validation for XSS attacks and what special characters the validation should consider, the next challenge is to identify how your validation handles special characters. If special characters are not considered valid input to the application, then you can reject any input that contains special characters as invalid. A second option is to remove special characters with filtering. However, filtering has the side effect of changing any visual representation of the filtered content and might be unacceptable in circumstances where the integrity of the input must be preserved for display.

If input containing special characters must be accepted and displayed accurately, validation must encode any special characters to remove their significance. A complete list of ISO 8859-1 encoded values for special characters is provided as part of the official HTML specification [2].

Many application servers attempt to limit an application's exposure to cross-site scripting vulnerabilities by providing implementations for the functions responsible for setting certain specific HTTP response content that perform validation for the characters essential to a cross-site scripting attack. Do not rely on the server running your application to make it secure. For any developed application, there are no guarantees about which application servers it will run on during its lifetime. As standards and known exploits evolve, there are no guarantees that application servers will continue to stay in sync.

Tips:

1. The Fortify Secure Coding Rulepacks warn about SQL Injection issues when untrusted data is written to a database and also treat the database as a source of untrusted data, which can lead to XSS vulnerabilities. If the database is a trusted resource in your environment, use custom filters to filter out dataflow issues that include the DATABASE taint flag or originate from database sources. Nonetheless, it is often still a good idea to validate everything read from the database.

2. Even though URL encoding untrusted data protects against many XSS attacks, some browsers automatically decode content at certain locations within the Document Object Model (DOM) prior to passing it to the JavaScript interpreter. To reflect this danger, the Rulepacks no longer treat URL encoding routines as sufficient to protect against cross-site scripting. Data values that are URL encoded and subsequently output will cause Fortify to report Cross-Site Scripting: Poor Validation vulnerabilities.

3. Older versions of React are more susceptible to cross-site scripting attacks by controlling an entire component. Newer versions use Symbols to identify a React component, which prevents the exploit, however older browsers that do not have Symbol support (natively, or through polyfills) are still vulnerable. Other types of cross-site scripting attacks are valid for all browsers and versions of React.

main.js, line 48 (Cross-Site Scripting: DOM)

Fortify Priority:	Critical	**Folder**	Critical
Kingdom:	Input Validation and Representation		

Abstract: The method addFilter() in main.js sends unvalidated data to a web browser on line 48, which can result in the browser executing malicious code.

Source: main.js:48 Read localStorage.currentTabUrl()

```
46      if (localStorage.currentTabUrl) {
47        const parser = document.createElement('a');
48        parser.href = localStorage.currentTabUrl;
49        urlRegex = parser.origin + '/.*';
50      }
```

Sink: main.js:48 Assignment to parser.href()

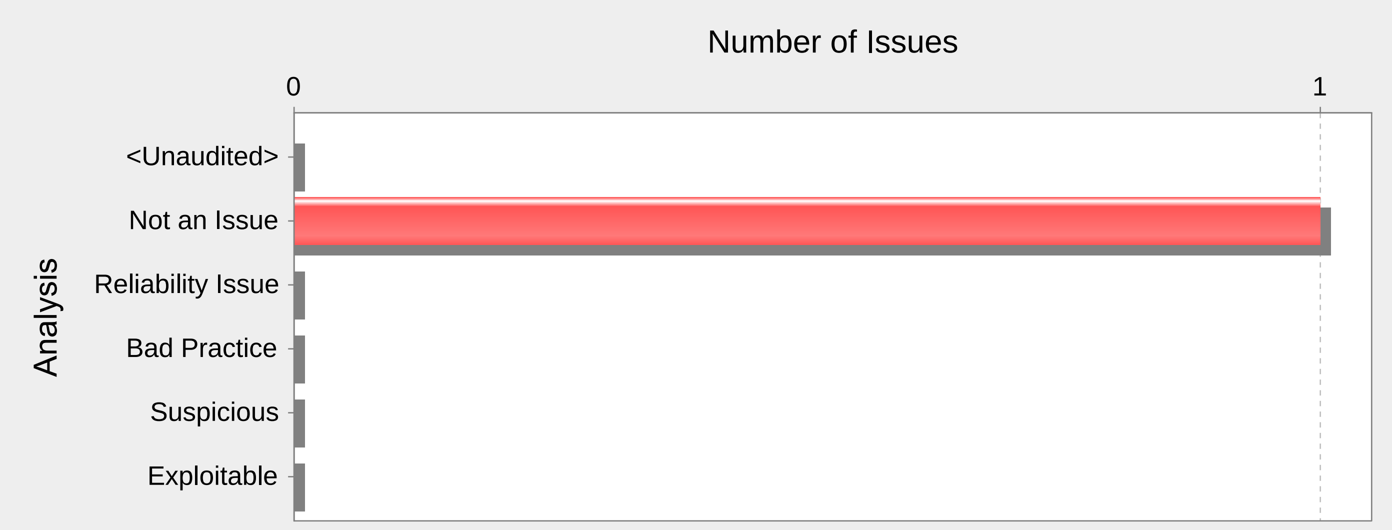
```
46      if (localStorage.currentTabUrl) {
47        const parser = document.createElement('a');
48        parser.href = localStorage.currentTabUrl;
49        urlRegex = parser.origin + '/.*';
50      }
```

Analysis: Not an Issue

Comments: *jpyeron 2024-03-05 11:01 AM* This is a A tag generation to have access to the origin string calculation from the current browser tab URL - this is used to later greate a REGEX patteren for matching "same site" when the UI has only this tab/site checked.

There are no user clickable URLs made.
There are no sources of injection.

Category: Insecure Randomness (1 Issues)



Abstract:

The random number generator implemented by random() cannot withstand a cryptographic attack.

Explanation:

Insecure randomness errors occur when a function that can produce predictable values is used as a source of randomness in a security-sensitive context.

Computers are deterministic machines, and as such are unable to produce true randomness. Pseudorandom Number Generators (PRNGs) approximate randomness algorithmically, starting with a seed from which subsequent values are calculated.

There are two types of PRNGs: statistical and cryptographic. Statistical PRNGs provide useful statistical properties, but their output is highly predictable and form an easy to reproduce numeric stream that is unsuitable for use in cases where security depends on generated values being unpredictable. Cryptographic PRNGs address this problem by generating output that is more difficult to predict. For a value to be cryptographically secure, it must be impossible or highly improbable for an attacker to distinguish between the generated random value and a truly random value. In general, if a PRNG algorithm is not advertised as being cryptographically secure, then it is probably a statistical PRNG and should not be used in security-sensitive contexts, where its use can lead to serious vulnerabilities such as easy-to-guess temporary passwords, predictable cryptographic keys, session hijacking, and DNS spoofing.

Example: The following code uses a statistical PRNG to create a URL for a receipt that remains active for some period of time after a purchase.

```
function genReceiptURL (baseURL){
var randNum = Math.random();
var receiptURL = baseURL + randNum + ".html";
return receiptURL;
}
```

This code uses the Math.random() function to generate "unique" identifiers for the receipt pages it generates. Since Math.random() is a statistical PRNG, it is easy for an attacker to guess the strings it generates. Although the underlying design of the receipt system is also faulty, it would be more secure if it used a random number generator that did not produce predictable receipt identifiers, such as a cryptographic PRNG.

Recommendations:

When unpredictability is critical, as is the case with most security-sensitive uses of randomness, use a cryptographic PRNG. Regardless of the PRNG you choose, always use a value with sufficient entropy to seed the algorithm. (Do not use values such as the current time because it offers only negligible entropy.)

In JavaScript, the typical recommendation is to use the window.crypto.random() function in the Mozilla API. However, this method does not work in many browsers, including more recent versions of Mozilla Firefox. There is currently no cross-browser solution for a robust cryptographic PRNG. In the meantime, consider handling any PRNG functionality outside of JavaScript.

main.js, line 573 (Insecure Randomness)

Fortify Priority:	High	**Folder**	High
Kingdom:	Security Features		

Abstract: The random number generator implemented by random() cannot withstand a cryptographic attack.

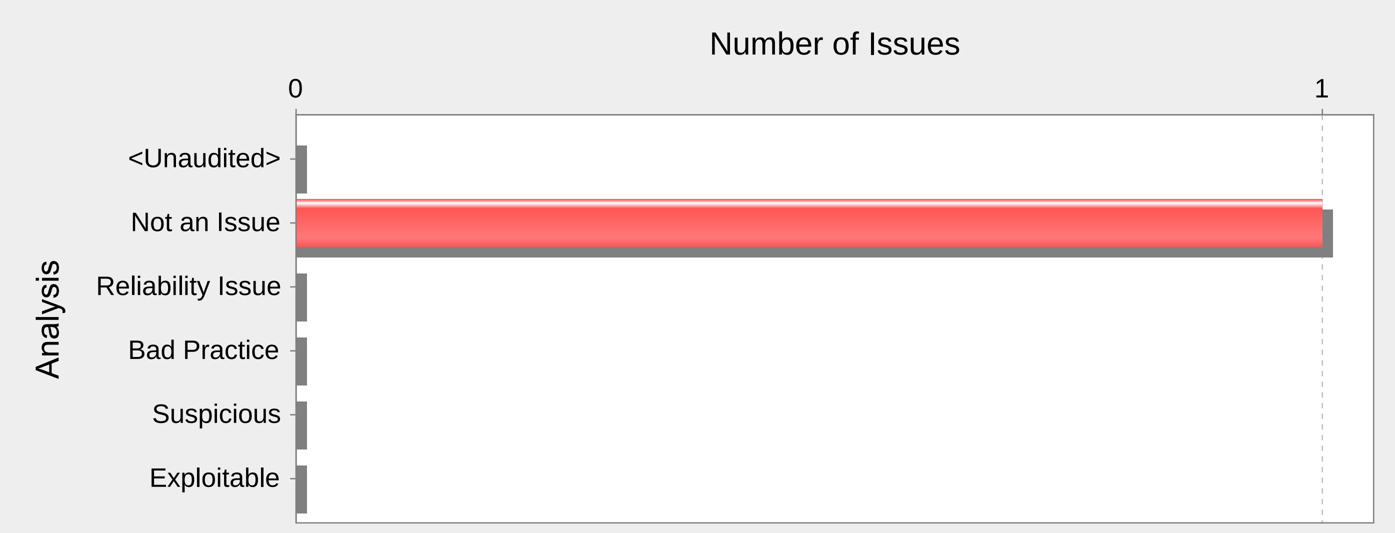
Sink: main.js:573 FunctionPointerCall: random()

```
571                {text: 'Tip: Pause button will temporarily pause all modifications'},
];
const tip = tips[Math.floor(Math.random() * tips.length)];
$mdToast.show({
  position: 'bottom',
```

Analysis: Not an Issue

Comments: *jpyeron 2024-03-05 11:00 AM* The random number is to selecta tip of the day, it has no security relevance.

Category: Open Redirect (1 Issues)



Abstract:

The file main.js passes unvalidated data to an HTTP redirect function on line 48. Allowing unvalidated input to control the URL used in a redirect can aid phishing attacks.

Explanation:

Redirects allow web applications to direct users to different pages within the same application or to external sites. Applications utilize redirects to aid in site navigation and, in some cases, to track how users exit the site. Open redirect vulnerabilities occur when a web application redirects clients to any arbitrary URL that can be controlled by an attacker.

Attackers might utilize open redirects to trick users into visiting a URL to a trusted site, but then redirecting them to a malicious site. By encoding the URL, an attacker can make it difficult for end-users to notice the malicious destination of the redirect, even when it is passed as a URL parameter to the trusted site. Open redirects are often abused as part of phishing scams to harvest sensitive end-user data.

Example 1: The following JavaScript code instructs the user's browser to open a URL read from the dest request parameter when a user clicks the link.

```
...
strDest = form.dest.value;
window.open(strDest,"myresults");
...
```

If a victim received an email instructing them to follow a link to "http://trusted.example.com/ecommerce/redirect.asp?dest=www.wilyhacker.com", the user would likely click on the link believing they would be transferred to the trusted site. However, when the victim clicks the link, the code in Example 1 will redirect the browser to "http://www.wilyhacker.com".

Many users have been educated to always inspect URLs they receive in emails to make sure the link specifies a trusted site they know. However, if the attacker Hex encoded the destination url as follows:

"http://trusted.example.com/ecommerce/redirect.asp?dest=%77%69%6C%79%68%61%63%6B%65%72%2E%63%6F%6D"

then even a savvy end-user may be fooled into following the link.

Recommendations:

Unvalidated user input should not be allowed to control the destination URL in a redirect. Instead, use a level of indirection: create a list of legitimate URLs that users are allowed to specify, and only allow users to select from the list. With this approach, input provided by users is never used directly to specify a URL for redirects.

Example 2: The following code references an array populated with valid URLs. The link the user clicks passes in the array index that corresponds to the desired URL.

```
...
strDest = form.dest.value;
if((strDest.value != null)||(strDest.value.length!=0))
{
if((strDest >= 0) && (strDest <= strURLArray.length -1 ))
{
strFinalURL = strURLArray[strDest];
window.open(strFinalURL,"myresults");
}
}
...
```

In some situations this approach is impractical because the set of legitimate URLs is too large or too hard to keep track of. In such cases, use a similar approach to restrict the domains that users can be redirected to, which can at least prevent attackers from sending users to malicious external sites.

main.js, line 48 (Open Redirect)

Fortify Priority:	Critical	**Folder**	Critical
Kingdom:	Input Validation and Representation		

Abstract: The file main.js passes unvalidated data to an HTTP redirect function on line 48. Allowing unvalidated input to control the URL used in a redirect can aid phishing attacks.

Source: main.js:48 Read localStorage.currentTabUrl()

```
46        if (localStorage.currentTabUrl) {
47          const parser = document.createElement('a');
48          parser.href = localStorage.currentTabUrl;
49          urlRegex = parser.origin + '/.*';
50        }
```

Sink: main.js:48 Assignment to parser.href()

```
46        if (localStorage.currentTabUrl) {
47          const parser = document.createElement('a');
48          parser.href = localStorage.currentTabUrl;
49          urlRegex = parser.origin + '/.*';
50        }
```

Analysis: Not an Issue

Comments: *jpyeron 2024-03-05 11:00 AM* not a redirect.

Detailed Project Summary

Files Scanned

Code base location: C:/projects/modheader

Files Scanned:

.git/config generic 10.6 KB Mar 4, 2024, 5:21:18 PM
.git/logs/HEAD generic   Mar 4, 2024, 5:21:53 PM
.git/logs/refs/heads/master generic   Mar 4, 2024, 5:21:53 PM
src/background.js typescript 219 Lines 12 KB Mar 4, 2024, 5:21:53 PM
src/cloudbackupdialog.tmpl.html html   Mar 4, 2024, 5:21:37 PM
src/exportdialog.tmpl.html html   Mar 4, 2024, 5:21:37 PM
src/footer.tmpl.html html   Mar 4, 2024, 5:21:37 PM
src/importdialog.tmpl.html html   Mar 4, 2024, 5:21:37 PM
src/manifest.json json 27 Lines Mar 4, 2024, 5:21:53 PM
src/popup.html html 14.2 KB Mar 4, 2024, 5:21:53 PM
src/scripts/angular-material.min.js typescript 104.8 KB Mar 4, 2024, 5:21:37 PM
src/scripts/main.js typescript 318 Lines 16.1 KB Mar 4, 2024, 5:21:53 PM
src/settings.tmpl.html html   Mar 4, 2024, 5:21:37 PM
src/styles/angular-material.min.css generic 97.3 KB Mar 4, 2024, 5:21:37 PM

Reference Elements

Classpath:

No classpath specified during translation

Libdirs:

No libdirs specified during translation

Rulepacks

Valid Rulepacks:

Name: Fortify Secure Coding Rules, Community, Cloud
Version: 2023.4.0.0006
ID: 686C4B2F-0321-4025-B9F4-6E26094B4746
SKU: RUL13242

Name: Fortify Secure Coding Rules, Community, Universal
Version: 2023.4.0.0006
ID: 97b8b0e6-618b-47cf-a7fb-8636faea6b75
SKU: RUL13240

Name: Fortify Secure Coding Rules, Core, Cloud
Version: 2023.4.0.0006
ID: BDACC98E-569C-4ECC-92AA-8DD890DF1287
SKU: RUL13249

Name: Fortify Secure Coding Rules, Core, JavaScript
Version: 2023.4.0.0006
ID: BD292C4E-4216-4DB8-96C7-9B607BFD9584

SKU: RUL13059

Name: Fortify Secure Coding Rules, Core, Universal
Version: 2023.4.0.0006
ID: 88D39959-D322-499A-87F3-BC9E1193B07A
SKU: RUL13241

Name: Fortify Secure Coding Rules, Extended, Configuration
Version: 2023.4.0.0006
ID: CD6959FC-0C37-45BE-9637-BAA43C3A4D56
SKU: RUL13005

Name: Fortify Secure Coding Rules, Extended, Content
Version: 2023.4.0.0006
ID: 9C48678C-09B6-474D-B86D-97EE94D38F17
SKU: RUL13067

Name: Fortify Secure Coding Rules, Extended, JavaScript
Version: 2023.4.0.0006
ID: C4D1969E-B734-47D3-87D4-73962C1D32E2
SKU: RUL13141

External Metadata:
Version: 2023.4.0.0006

Name: CIS AKS 1.3.0
ID: E244A68A-528E-4ADA-A935-B0EC0BD7109F
The CIS Azure Kubernetes Service (AKS) Benchmark v1.3.0 from the Center for Internet Security (CIS) provides prescriptive guidance for establishing a secure baseline configuration for Azure Kubernetes Service.

Name: CIS Amazon Web Services 2.0.0
ID: F7D454BE-9612-4D3B-9A11-08886232FB2D
The CIS Amazon Web Services Foundations Benchmark v2.0.0 from the Center for Internet Security (CIS) provides prescriptive guidance for establishing a secure baseline configuration for Amazon Web Services.

Name: CIS EKS 1.3.0
ID: A5FCC3D2-3F76-49A6-9978-545D7E85EC55
The CIS Amazon Elastic Kubernetes Service (EKS) Benchmark v1.3.0 from the Center for Internet Security (CIS) provides prescriptive guidance for establishing a secure baseline configuration for Amazon Elastic Kubernetes Service.

Name: CIS GKE 1.4.0
ID: 800E87DB-CC72-48B1-8AB1-5C4D30B964F0
The CIS Google Kubernetes Engine (GKE) Benchmark v1.4.0 from the Center for Internet Security (CIS) provides prescriptive guidance for establishing a secure baseline configuration for Google Kubernetes Engine Service.

Name: CIS Google Cloud 2.0.0
ID: 0295D43A-6909-4EEC-B939-1244B2905862
The CIS Google Cloud Platform Foundation Benchmark v2.0.0 from the Center for Internet Security (CIS) provides prescriptive guidance for establishing a secure baseline configuration for Google Cloud.

Name: CIS Kubernetes 1.7.1

ID: F5200B7A-129F-482B-BF6C-AB1EBD21553E

The CIS Kubernetes Benchmark v1.7.1 from the Center for Internet Security (CIS provides prescriptive guidance for establishing a secure baseline configuration for Kubernetes v1.25.

Name: CIS Microsoft Azure 2.0.0

ID: E57A2F1A-B80F-4DA1-99A7-68BECDC54891

The CIS Microsoft Azure Foundations Benchmark from the Center for Internet Security (CIS) provides prescriptive guidance for establishing a secure baseline configuration for Microsoft Azure.

Name: CWE

ID: 3ADB9EE4-5761-4289-8BD3-CBFCC593EBBC

The Common Weakness Enumeration (CWE), co-sponsored and maintained by MITRE, is international in scope and free for public use. CWE provides a unified, measurable set of software weaknesses that is enabling more effective discussion, description, selection, and use of software security tools and services that can find these weaknesses in source code and operational systems as well as better understanding and management of software weaknesses related to architecture and design.

Name: CWE Top 25 2021

ID: FDA85EBD-56E5-4698-86FD-DD52E2F8F32B

The 2021 CWE Top 25 Most Dangerous Software Errors lists the most widespread and critical weaknesses that can lead to serious vulnerabilities in software (as demonstrated by the National Vulnerability Database). These weaknesses occur frequently, are often easy to find, and easy to exploit. They are dangerous because they will frequently enable attackers to completely take over the software, steal data, or prevent the software from working at all. The list is the result of heuristic formula that the CWE Team used with a data-driven approach that leveraged the Common Vulnerabilities and Exposure (CVE), National Vulnerability Database (NVD), and Common Vulnerability Scoring System (CVSS). Due to the hierarchical nature of the CWE taxonomy, Fortify considers all CWE IDs which are children of a Top 25 entry, as included within the context of the entry due to the "CHILD-OF" relationship within the hierarchy. Exercise caution if using only this Top 25 list to prioritize auditing efforts because the software under analysis might not align with the assumptions of the heuristic used to define the Top 25. For example, many of these weaknesses are related to C-like languages and the software under analysis might not be within the C-family of languages - thus, many CWEs would not be in scope.

Name: CWE Top 25 2022

ID: D16E16F3-91AE-4AA3-A943-FCDE765446E5

The 2022 CWE Top 25 Most Dangerous Software Errors lists the most widespread and critical weaknesses that can lead to serious vulnerabilities in software (as demonstrated by the National Vulnerability Database). These weaknesses occur frequently, are often easy to find, and easy to exploit. They are dangerous because they will frequently enable attackers to completely take over the software, steal data, or prevent the software from working at all. The list is the result of heuristic formula that the CWE Team used with a data-driven approach that leveraged the Common Vulnerabilities and Exposure (CVE), National Vulnerability Database (NVD), and Common Vulnerability Scoring System (CVSS). Due to the hierarchical nature of the CWE taxonomy, Fortify considers all CWE IDs which are children of a Top 25 entry, as included within the context of the entry due to the "CHILD-OF" relationship within the hierarchy. Exercise caution if using only this Top 25 list to prioritize auditing efforts because the software under analysis might not align with the assumptions of the heuristic used to define the Top 25. For example, many of these weaknesses are related to C-like languages and the software under analysis might not be within the C-family of languages - thus, many CWEs would not be in scope.

Name: CWE Top 25 2023

ID: D0A4A1E9-36C4-4E47-9F0F-597D22E6DC66

The 2023 CWE Top 25 Most Dangerous Software Errors lists the most widespread and critical weaknesses that can lead to serious vulnerabilities in software (as demonstrated by the National Vulnerability Database). These weaknesses occur frequently, are often easy to find, and easy to exploit. They are dangerous because they will frequently enable attackers to completely take over the software, steal data, or prevent the software from working at all. The list is the result of heuristic formula that the CWE Team used with a data-driven approach that leveraged the Common Vulnerabilities and Exposure (CVE), National Vulnerability

Database (NVD), and Common Vulnerability Scoring System (CVSS). Due to the hierarchical nature of the CWE taxonomy, Fortify considers all CWE IDs which are children of a Top 25 entry, as included within the context of the entry due to the "CHILD-OF" relationship within the hierarchy. Exercise caution if using only this Top 25 list to prioritize auditing efforts because the software under analysis might not align with the assumptions of the heuristic used to define the Top 25. For example, many of these weaknesses are related to C-like languages and the software under analysis might not be within the C-family of languages - thus, many CWEs would not be in scope.

Name: DISA CCI 2
ID: 7F037130-41E5-40F0-B653-7819A4B3E241
The purpose of a Defense Information Systems Agency (DISA) Control Correlation Identifier (CCI) is to provide a standard identifier for policy based requirements which connect high-level policy expressions and low-level technical implementations. Associated with each CCI is a description for each of the singular, actionable, statements compromising an information assurance (IA) control or IA best practice. Using CCI allows high-level policy framework security requirements to be decomposed and explicitly associated with low-level implementations, thus enabling the assessment of related compliance assessment results spanning heterogeneous technologies. The current IA controls and best practices associated with each CCI, that are specified in NIST SP 800-53 Revision 4, can be viewed using the DISA STIG Viewer.
The following table summarizes the number of issues identified across the different CCIs broken down by Fortify Priority Order. The status of a CCI is considered "In Place" when there are no issues reported for a given CCI.

If the project is missing a Fortify Static Code Analyzer (SCA) scan, or the scan contains findings that have not been fixed, hidden or suppressed, CCI-003187 is not considered "In Place". Similarly, if the project is missing a Fortify WebInspect scan, or the scan contains any critical findings, CCI-000366 and CCI-000256 are not considered "In Place".

Name: FISMA
ID: B40F9EE0-3824-4879-B9FE-7A789C89307C
The Federal Information Processing Standard (FIPS) 200 document is part of the official series of publications, issued by the National Institute of Standards and Technology (NIST), relating to standards and guidelines adopted and promulgated under the provisions of the Federal Information Security Management Act (FISMA). Specifically, FIPS Publication 200 specifies the "Minimum Security Requirements for Federal Information and Information Systems."

Name: GDPR
ID: 771C470C-9274-4580-8556-C12F5E4BEC51
The EU General Data Protection Regulation (GDPR) replaces the Data Protection Directive 95/46/EC and was designed to harmonize data privacy laws across Europe, to protect and empower all EU citizens data privacy and to reshape the way organizations across the region approach data privacy. Going into effect on May 25, 2018, GDPR provides a framework for organizations on how to handle personal data. According to GDPR regulation personal data "means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person." GDPR articles that pertain to application security and require businesses to protect personal data during design and development of its product and services are:

- Article 25, Data protection by design and by default - which requires "The controller shall implement appropriate technical and organisational measures for ensuring that, by default, only personal data which are necessary for each specific purpose of the processing are processed."
- Article 32, Security of processing - which requires businesses to protect its systems and applications "from accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access to personal data". This report may be used by organizations as a framework to help identify and protect personal data as it relates to application security.

Name: MISRA C 2012
ID: 555A3A66-A0E1-47AF-910C-3F19A6FB2506

Now in its third edition, the Motor Industry Software Reliability Association (MISRA) C Guidelines describe a subset of the C programming language in which there is reduced risk of introducing mistakes in critical systems. While the MISRA C Guidelines focus upon safety-related software development, a subset of the rules reflects security properties. Fortify interprets the MISRA C Guidelines under the context of security and provides correlation of security vulnerability categories to the rules defined by MISRA. Fortify provides these security focused detection mechanisms with the standard rulepacks, however, further support of the MISRA C Guidelines related to safety can be added through the use of custom rules. The results in this report can assist in the creation of a compliance matrix for MISRA.

Name: MISRA C++ 2008
ID: 5D4B75A1-FC91-4B4B-BD4D-C81BBE9604FA
The Motor Industry Software Reliability Association (MISRA) C++ Guidelines describe a subset of the C++ programming language in which there is reduced risk of introducing mistakes in critical systems. While the MISRA C++ Guidelines focus upon safety-related software development, a subset of the rules reflects security properties. Fortify interprets the MISRA C++ Guidelines under the context of security and provides correlation of security vulnerability categories to the rules defined by MISRA. Fortify provides these security focused detection mechanisms with the standard rulepacks, however, further support of the MISRA C++ Guidelines related to safety can be added through the use of custom rules. The results in this report can assist in the creation of a compliance matrix for MISRA.

Name: NIST SP 800-53 Rev.4
ID: 1114583B-EA24-45BE-B7F8-B61201BACDD0
NIST Special Publication 800-53 Revision 4 provides a list of security and privacy controls designed to protect federal organizations and information systems from security threats. The following table summarizes the number of issues identified across the different controls and broken down by Fortify Priority Order.

Name: NIST SP 800-53 Rev.5
ID: 32434089-54F3-49F8-93F8-688B6B2FE8ED
NIST Special Publication 800-53 Revision 5 provides a list of security and privacy controls designed to protect federal organizations and information systems from security threats. The following table summarizes the number of issues identified across the different controls and broken down by Fortify Priority Order.

Name: OWASP API Top 10 2023
ID: 5240B1AE-EB36-4DBD-9FC3-34BC60F05EE0
The OWASP API Top 10 2023 provides a list of the top security risks affecting APIs in 2023. It aims to raise awareness around API security weaknesses and to educate those involved in API development and maintenance, such as developers, designers, architects, managers and/or organizations in general who need to secure Web APIs. The OWASP API Top 10 focuses on weaknesses affecting Web APIs and it is not intended to be used only by itself, instead it is intended to be used in combination with other standards and best practices in order to thoroughly capture all relevant risks. For example: it should be used in combination with the OWASP Top 10 in order to identify issues related to input validation such as injections.

Name: OWASP ASVS 4.0
ID: 28083E33-760F-4A1A-AADA-738CC60082AD
The OWASP Application Security Verification Standard establishes a framework of security requirements and controls that focus on functional and non-functional security controls for the software development lifecycle based upon a community-driven effort. OWASP ASVS identifies several application security verification levels, with each level increasing depth:

<LI>ASVS Level 1 (L1): for low assurance levels and is completely penetration testable.</LI>
<LI>ASVS Level 2 (L2): for applications that contain sensitive data, which requires protection, and is the recommended level for most apps.</LI>
<LI>ASVS Level 3 (L3): for the most critical applications - applications that perform high value transactions, contain sensitive medical data, or any application that requires the highest level of trust.</LI> </UL>

Name: OWASP MASVS 2.0
ID: 89717E5C-2EC1-4D41-9A7C-6BCD91C8E9AA
The OWASP Mobile Application Security Verification Standard (MASVS) 2.0 provides a powerful awareness document for mobile application security. The OWASP MASVS represents a broad consensus about what the most critical mobile application security flaws are. MASVS is intended to be used by architects, and developers, aiming to develop secure mobile applications. Testers evaluating mobile applications for completeness of security control coverage, and mitigation of risks can also use MASVS as a guide in addition to the OWASP Mobile Security Testing Guide (MASTG). Mobile apps often interact with diverse systems such as backend servers, third-party APIs, Bluetooth devices, cars, and IoT devices, each introducing its own set of security risks. Consequently, these risks must be considered during the mobile app's security design and threat modeling process. For instance, when engaging with a backend server, it is essential to employ the OWASP Application Security Verification Standard (ASVS), in addition to the OWASP MASVS, to ensure the server meets the required security standards.

Name: OWASP Mobile 2014
ID: EEE3F9E7-28D6-4456-8761-3DA56C36F4EE
The OWASP Mobile Top 10 Risks 2014 provides a powerful awareness document for mobile application security. The OWASP Mobile Top 10 represents a broad consensus about what the most critical mobile application security flaws are. Project members include a variety of security experts from around the world who have shared their expertise to produce this list.

Name: OWASP Mobile Top 10 2023
ID: 8597362A-CD58-47EA-B450-890DDEE7D545
The OWASP Mobile Top 10 Risks 2023 provides a list of the top 10 risks affecting mobile applications in 2023. It represents a broad consensus on what the most critical mobile application security vulnerabilities are in 2023. The OWASP Mobile Top 10 focuses on weaknesses affecting mobile applications and it is not intended to be used only by itself. Instead it is intended to be used in combination with other standards and best practices in order to thoroughly capture all relevant risks. For example, it should be used in combination with the OWASP API Top 10 in order to identify issues related to requests sent from the mobile application to the server.

Name: OWASP Top 10 2013
ID: 1A2B4C7E-93B0-4502-878A-9BE40D2A25C4
The OWASP Top Ten 2013 provides a powerful awareness document for web application security. The OWASP Top Ten represents a broad consensus about what the most critical web application security flaws are. Project members include a variety of security experts from around the world who have shared their expertise to produce this list.

Name: OWASP Top 10 2017
ID: 3C6ECB67-BBD9-4259-A8DB-B49328927248
The OWASP Top Ten 2017 provides a powerful awareness document for web application security focused on informing the community about the consequences of the most common and most important web application security weaknesses. The OWASP Top Ten represents a broad agreement about what the most critical web application security flaws are with consensus being drawn from data collection and survey results. Project members include a variety of security experts from around the world who have shared their expertise to produce this list.

Name: OWASP Top 10 2021
ID: 1887A283-3C0D-453C-AD10-0B451EAF096D0
The OWASP Top 10 2021 provides a powerful awareness document for web application security focused on informing the community about the consequences of the most common and most important web application security weaknesses. The OWASP Top 10 represents a broad agreement about what the most critical web application security flaws are with consensus drawn from data collection and survey results. Project members include a variety of security experts from around the world who have shared their expertise to produce this list.

Name: PCI 3.2.1
ID: EADE255F-6561-4EFE-AD31-2914F6BFA329
The following is a summary of the application security portions of Payment Card Industry (PCI) Data Security Standard (DSS) v3.2.1. Fortify tests for 31 application security related requirements across sections 1, 2, 3, 4, 6, 7, 8, and 10 of PCI DSS and reports whether each requirement is In Place or Not In Place to indicate whether requirements are satisfied or not. This report is intended to measure the level of adherence the specific application(s) possess when compared to PCI DSS 3.2.1 compliance and is not intended to serve as a comprehensive Report on Compliance (ROC). The information contained in this report is targeted at project managers, security auditors, and compliance auditors.

Name: PCI 4.0
ID: A2937951-C21A-4CFF-9A73-220C67EBDD8C
The following is a summary of the application security portions of Payment Card Industry (PCI) Data Security Standard (DSS) v4.0. Fortify tests for 39 application security related requirements across sections 1, 2, 3, 4, 5, 6, 7, 8, 10 and 12 of PCI DSS and reports whether each requirement is In Place or Not In Place to indicate whether requirements are satisfied or not. This report is intended to measure the level of adherence the specific application(s) possess when compared to PCI DSS 4.0 compliance and is not intended to serve as a comprehensive Report on Compliance (ROC). The information contained in this report is targeted at project managers, security auditors, and compliance auditors.

Name: PCI SSF 1.0
ID: 0F551543-AF0E-4334-BEDF-1DDCD5F4BF74
The following is a summary of the application security portions of the Secure Software Requirements and Assessment Procedures defined in the Payment Card Industry (PCI) Software Security Framework (SSF) v1.0. Fortify tests for 23 application security related control objectives across Control Objective sections 2, 3, 4, 5, 6, 7, 8, and A.2 of PCI SSF and reports whether each control objective is In Place or Not In Place to indicate whether requirements are satisfied or not. This report is intended to measure the level of adherence the specific application(s) possess when compared to PCI SSF 1.0 compliance and is not intended to serve as a comprehensive Report on Compliance (ROC). The information contained in this report is targeted at project managers, security auditors, and compliance auditors.

Name: PCI SSF 1.1
ID: 601EA2F3-5EDC-411C-818C-10DC5B29467D
The following is a summary of the application security portions of the Secure Software Requirements and Assessment Procedures defined in the Payment Card Industry (PCI) Software Security Framework (SSF) v1.1. Fortify tests for 31 application security related control objectives across Control Objective sections 2, 3, 4, 5, 6, 7, 8, A.2, B.2, and B.3 of PCI SSF and reports whether each control objective is In Place or Not In Place to indicate whether requirements are satisfied or not. This report is intended to measure the level of adherence the specific application(s) possess when compared to PCI SSF 1.1 compliance and is not intended to serve as a comprehensive Report on Compliance (ROC). The information contained in this report is targeted at project managers, security auditors, and compliance auditors.

Name: PCI SSF 1.2
ID: 184D9BE1-8DE8-4322-92D2-3BDCC9148241
The following is a summary of the application security portions of the Secure Software Requirements and Assessment Procedures defined in the Payment Card Industry (PCI) Software Security Framework (SSF) v1.2. Fortify tests for 44 application security related control objectives across Control Objective sections 2, 3, 4, 5, 6, 7, 8, 10, A.2, B.2, B.3, C.1, C.2, C.3 and C.4 of PCI SSF and reports whether each control objective is In Place or Not In Place to indicate whether requirements are satisfied or not. This report is intended to measure the level of adherence the specific application(s) possess when compared to PCI SSF 1.2 compliance and is not intended to serve as a comprehensive Report on Compliance (ROC). The information contained in this report is targeted at project managers, security auditors, and compliance auditors.

Name: SANS Top 25 2011
ID: 92EB4481-1FD9-4165-8E16-F2DE6CB0BD63
SANS Top 25 2011 Most Dangerous Software Errors provides an enumeration of the most widespread and critical errors,

categorized by Common Weakness Enumeration (CWE) identifiers, that lead to serious vulnerabilities in software (http://cwe.mitre.org/). These software errors are often easy to find and exploit. The inherent danger in these errors is that they can allow an attacker to completely take over the software, steal data, or prevent the software from working at all.

Name: STIG 4.11
ID: D9F6C005-1ED5-4685-8A69-79A87A1A9431
Each requirement or recommendation identified by the Defense Information Systems Agency (DISA) STIG is represented by a STIG identifier (STIGID), which corresponds to a checklist item and a severity code [APSC-DV-<I>ID</I>: CAT <I>SEV</I>]. DISA STIG defines three severities with respect to vulnerabilities where their:

<LI>exploitation leads to direct and immediate loss of Confidentiality, Availability, or Integrity (CAT I).</LI>
<LI>exploitation potentially results in loss of Confidentiality, Availability, or Integrity (CAT II).</LI>
<LI>existence degrades protections against loss of Confidentiality, Availability, or Integrity (CAT III).</LI> </UL>

The following table summarizes the number of issues identified across the different STIGIDs broken down by Fortify Priority Order. The status of a STIGID is considered "In Place" when there are no issues reported for a given STIGID.

If the project is missing a Fortify Static Code Analyzer (SCA) scan, or the scan contains findings that have not been fixed, hidden or suppressed, STIGID APSC-DV-003170: CAT II is not considered "In Place". Similarly, if the project is missing a Fortify WebInspect scan, or the scan contains any critical findings, STIGID APSC-DV-001460: CAT II and STIGID APSC-DV-002930: CAT II are not considered "In Place".

Name: STIG 5.1
ID: 1E2530B5-61C5-45D0-B479-79CB82DAFF83
Each requirement or recommendation identified by the Defense Information Systems Agency (DISA) STIG is represented by a STIG identifier (STIGID), which corresponds to a checklist item and a severity code [APSC-DV-<I>ID</I>: CAT <I>SEV</I>]. DISA STIG defines three severities with respect to vulnerabilities where their:

<LI>exploitation leads to direct and immediate loss of Confidentiality, Availability, or Integrity (CAT I).</LI>
<LI>exploitation potentially results in loss of Confidentiality, Availability, or Integrity (CAT II).</LI>
<LI>existence degrades protections against loss of Confidentiality, Availability, or Integrity (CAT III).</LI> </UL>

The following table summarizes the number of issues identified across the different STIGIDs broken down by Fortify Priority Order. The status of a STIGID is considered "In Place" when there are no issues reported for a given STIGID.

If the project is missing a Fortify Static Code Analyzer (SCA) scan, or the scan contains findings that have not been fixed, hidden or suppressed, STIGID APSC-DV-003170: CAT II is not considered "In Place". Similarly, if the project is missing a Fortify WebInspect scan, or the scan contains any critical findings, STIGID APSC-DV-001460: CAT II and STIGID APSC-DV-002930: CAT II are not considered "In Place".

Name: STIG 5.2
ID: C264713B-7F56-4CE9-B77E-7240C7A2817E
Each requirement or recommendation identified by the Defense Information Systems Agency (DISA) STIG is represented by a STIG identifier (STIGID), which corresponds to a checklist item and a severity code [APSC-DV-<I>ID</I>: CAT <I>SEV</I>]. DISA STIG defines three severities with respect to vulnerabilities where their:

<LI>exploitation leads to direct and immediate loss of Confidentiality, Availability, or Integrity (CAT I).</LI>
<LI>exploitation potentially results in loss of Confidentiality, Availability, or Integrity (CAT II).</LI>
<LI>existence degrades protections against loss of Confidentiality, Availability, or Integrity (CAT III).</LI> </UL>

The following table summarizes the number of issues identified across the different STIGIDs broken down by Fortify Priority

Order. The status of a STIGID is considered "In Place" when there are no issues reported for a given STIGID.

If the project is missing a Fortify Static Code Analyzer (SCA) scan, or the scan contains findings that have not been fixed, hidden or suppressed, STIGID APSC-DV-003170: CAT II is not considered "In Place". Similarly, if the project is missing a Fortify WebInspect scan, or the scan contains any critical findings, STIGID APSC-DV-001460: CAT II and STIGID APSC-DV-002930: CAT II are not considered "In Place".

Name: STIG 5.3
ID: DA04684C-0AC8-44E8-BF65-C6CCD8B369C7
Each requirement or recommendation identified by the Defense Information Systems Agency (DISA) STIG is represented by a STIG identifier (STIGID), which corresponds to a checklist item and a severity code [APSC-DV-<I>ID</I>: CAT <I>SEV</I>]. DISA STIG defines three severities with respect to vulnerabilities where their:

<LI>exploitation leads to direct and immediate loss of Confidentiality, Availability, or Integrity (CAT I).</LI>
<LI>exploitation potentially results in loss of Confidentiality, Availability, or Integrity (CAT II).</LI>
<LI>existence degrades protections against loss of Confidentiality, Availability, or Integrity (CAT III).</LI> </UL>

The following table summarizes the number of issues identified across the different STIGIDs broken down by Fortify Priority Order. The status of a STIGID is considered "In Place" when there are no issues reported for a given STIGID.

If the project is missing a Fortify Static Code Analyzer (SCA) scan, or the scan contains findings that have not been fixed, hidden or suppressed, STIGID APSC-DV-003170: CAT II is not considered "In Place". Similarly, if the project is missing a Fortify WebInspect scan, or the scan contains any critical findings, STIGID APSC-DV-001460: CAT II and STIGID APSC-DV-002930: CAT II are not considered "In Place".

Name: SWC
ID: 834F2488-AE1E-4465-8E69-4BA5A54C37D0
The Smart Contract Weakness Classification (SWC) is a systematic framework that categorizes and explains vulnerabilities in smart contracts. It provides a standardized way to understand and address weaknesses in these self-executing code pieces running on blockchains like Ethereum. Notably, the SWC registry's content has not been comprehensively updated since 2020, resulting in known incompleteness, errors, and important omissions.

Name: WASC 2.00
ID: 74f8081d-dd49-49da-880f-6830cebe9777
The Web Application Security Consortium (WASC) was created as a cooperative effort to standardize, clarify, and organize the threats to the security of a web site. Version 2.00 of their Threat Classification outlines the attacks and weaknesses that can commonly lead to a website being compromised.

Properties

```
CA.EnableFineGrainedMergingProfiling=true
CA.EnableMachineProfiling=false
CA.EnableMergingProfiling=true
WinForms.CollectionMutationMonitor.Label=WinFormsDataSource
ast.loading.filter=false
awt.toolkit=sun.awt.windows.WToolkit
com.fortify.AuthenticationKey=C:\Users\jpyeron\AppData\Local/Fortify/config/tools
com.fortify.Core=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core
com.fortify.InstallRoot=C:\Program Files\Fortify\Fortify_SCA_23.1.0
com.fortify.InstallationUserName=jpyeron
com.fortify.SCAExecutablePath=C:/Program Files/Fortify/Fortify_SCA_23.1.0/bin/sourceanalyzer.exe
com.fortify.TotalPhysicalMemory=68388900864
com.fortify.VS.RequireASPPrecompilation=true
com.fortify.WorkingDirectory=C:\Users\jpyeron\AppData\Local/Fortify
com.fortify.locale=en
com.fortify.log.console=false
com.fortify.sca.APEXv2=false
com.fortify.sca.AddImpliedMethods=true
com.fortify.sca.AntCompilerClass=com.fortify.dev.ant.SCACompiler
com.fortify.sca.ApexVersion=57
com.fortify.sca.AppendLogFile=true
com.fortify.sca.AspnetTranslatorDotnet=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core/private-bin/sca/dotnet/aspnet-translator/Dotnet/aspcodegen.exe
com.fortify.sca.AspnetTranslatorFramework=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core/private-bin/sca/dotnet/aspnet-translator/Framework/aspcodegen.exe
com.fortify.sca.BuildID=modheader
com.fortify.sca.BuildOptions=-pid-file C:\Users\jpyeron\AppData\Local\Temp\PID1248118855053723705.tmp -b modheader -machine-output C:\projects\modheader
com.fortify.sca.BundleControlflowIssues=true
com.fortify.sca.CollectPerformanceData=true
com.fortify.sca.CustomRulesDir=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core\config\customrules
com.fortify.sca.DaemonCompilers=com.fortify.sca.util.compilers.GppCompiler,com.fortify.sca.util.compilers.GccCompiler,com.fortify.sca.util.compilers.AppleGppCompiler,com.fortify.sca.util.compilers.AppleGccCompiler,com.fortify.sca.util.compilers.MicrosoftCompiler,com.fortify.sca.util.compilers.MicrosoftLinker,com.fortify.sca.util.compilers.LdCompiler,com.fortify.sca.util.compilers.ArUtil,com.fortify.sca.util.compilers.SunCCompiler,com.fortify.sca.util.compilers.SunCppCompiler,com.fortify.sca.util.compilers.IntelCompiler,com.fortify.sca.util.compilers.ExternalCppAdapter,com.fortify.sca.util.compilers.ClangCompiler
com.fortify.sca.DeadCodeFilter=true
com.fortify.sca.DeadCodeIgnoreTrivialPredicates=true
com.fortify.sca.DefaultAnalyzers=semantic:dataflow:controlflow:nullptr:configuration:content:structural:buffer
com.fortify.sca.DefaultFileTypes=java,rb,erb,jsp,jspx,jspf,tag,tagx,tld,sql,cfm,php,phtml,ctp,pks,pkh,pkb,xml,config,Config,settings,properties,dll,exe,winmd,cs,vb,asax,ascx,ashx,asmx,aspx,master,Master,xaml,baml,cshtml,vbhtml,razor,inc,asp,vbscript,js,mjs,cjs,jsx,ini,bas,cls,vbs,frm,ctl,html,htm,xsd,wsdd,xmi,py,cfml,cfc,abap,xhtml,cpx,xcfg,jsff,as,mxml,cbl,cob,cscfg,csdef,wadcfg,wadcfgx,appxmanifest,wsdl,plist,bsp,ABAP,BSP,swift,page,trigger,scala,ts,mts,cts,tsx,conf,json,yaml,yml,tf,hcl,go,kt,kts,Dockerfile,dockerfile,vue
com.fortify.sca.DefaultJarsDirs=default_jars
com.fortify.sca.DefaultRulesDir=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core\config\rules
com.fortify.sca.DisableCFRules=19EF0414-88CD-4882-82FC-BF3A89865666,4E28CEFE-1B94-4711-BF5A-EDA5D1B3E6BF,A2D33B21-FE55-4C53-86C6-2AB5BF343738,7F4CC818-7525-440B-9C68-02267A80179A,7F80BA1C-82E9-4F2A-BBB4-ADFDFB27B215,E650C773-2BB6-42AA-BC29-370AAF0C53ED
com.fortify.sca.DisableDeadCodeElimination=false
com.fortify.sca.DisableFunctionPointers=false
com.fortify.sca.DisableGlobals=false
com.fortify.sca.DisableInferredConstants=false
com.fortify.sca.DotnetDecompiler=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core/private-bin/sca/dotnet/dotnet-decompiler/dotnet-decompiler.exe
com.fortify.sca.DotnetTranslator=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core/private-bin/sca/dotnet/dotnet-translator/dotnet-translator.exe
com.fortify.sca.EnableInterproceduralConstantResolution=true
com.fortify.sca.EnableNestedWrappers=true
com.fortify.sca.EnableStructuralMatchCache=true
com.fortify.sca.EnableWrapperDetection=true
com.fortify.sca.FVDLDisableDescriptions=false
com.fortify.sca.FVDLDisableProgramData=false
com.fortify.sca.FVDLDisableSnippets=false
com.fortify.sca.FVDLStylesheet=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core/resources/sca/fvdl2html.xsl
com.fortify.sca.GoTranslator=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core/private-bin/sca/golang/golang.exe
com.fortify.sca.IndirectCallGraphBuilders=WinFormsAdHocFunctionBuilder,VirtualCGBuilder,J2EEIndirectCGBuilder,JNICGBuilder,StoredProcedureResolver,JavaWSCGBuilder,StrutsCGBuilder,DotNetWSCGBuilder,SqlServerSPResolver,ASPCGBuilder,ScriptedCGBuilder,NewJspCustomTagCGBuilder,DotNetCABCGBuilder,StateInjectionCGBuilder,SqlServerSPResolver2,PHPLambdaResolver,JavaWebCGBuilder
com.fortify.sca.JVMArgs=-XX:+UseParallelGC -XX:SoftRefLRUPolicyMSPerMB=3000 --illegal-access=permit --add-exports=jdk.management/com.sun.management.internal=ALL-UNNAMED --add-exports=jdk.scripting.nashorn/jdk.nashorn.internal.runtime=ALL-UNNAMED --add-exports=java.base/jdk.internal.misc=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/sun.security.jca=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.util.regex=ALL-UNNAMED --add-opens=java.base/java.net=ALL-UNNAMED --add-opens=java.base/javax.crypto=ALL-UNNAMED --add-opens=java.management/sun.management=ALL-UNNAMED -Xmx35010M -Xss16M
com.fortify.sca.JavaSourcepathSearch=true
com.fortify.sca.JdkVersion=1.8
com.fortify.sca.LogFile=C:\Users\jpyeron\AppData\Local\Fortify\sca23.1\log\sca
com.fortify.sca.LogFileDir=C:\Users\jpyeron\AppData\Local\Fortify\sca23.1\log
com.fortify.sca.LogFileExt=.log
com.fortify.sca.LogFileName=sca.log
com.fortify.sca.LogFileNameNoExt=sca
com.fortify.sca.LogFilePath=C:\Users\jpyeron\AppData\Local\Fortify\sca23.1\log\sca.log
com.fortify.sca.LogLevel=INFO
com.fortify.sca.LowSeverityCutoff=1.0
com.fortify.sca.MachineOutputMode=
com.fortify.sca.MultithreadedAnalysis=true
com.fortify.sca.NoNestedOutTagOutput=org.apache.taglibs.standard.tag.rt.core.RemoveTag,org.apache.taglibs.standard.tag.rt.core.SetTag
com.fortify.sca.OldVbNetExcludeFileTypes=vb,asax,ascx,ashx,asmx,aspx,xaml,cshtml,vbhtml
com.fortify.sca.PHPVersion=7.4
com.fortify.sca.PHPv2=false
com.fortify.sca.PID=33416
com.fortify.sca.Phase0HigherOrder.Languages=python,ruby,swift,javascript,typescript
com.fortify.sca.Phase0HigherOrder.Level=1
com.fortify.sca.PidFile=C:\Users\jpyeron\AppData\Local\Temp\PID6845777546103778301.tmp
com.fortify.sca.PrintPerformanceDataAfterScan=false
com.fortify.sca.ProjectRoot=C:\Users\jpyeron\AppData\Local/Fortify
com.fortify.sca.ProjectRoot=C:\Users\jpyeron\AppData\Local/Fortify
com.fortify.sca.PythonV2=false
com.fortify.sca.PythonVersion=2
com.fortify.sca.Renderer=fpr
com.fortify.sca.RequireMapKeys=jsp_static
com.fortify.sca.ResultsFile=C:\Users\jpyeron\AppData\Local/Fortify\AWB-23.1.0\modheader\modheader.fpr
com.fortify.sca.ScaMSBuildDotnetTargets=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core/private-bin/sca/MSBuildPlugin/Dotnet/Fortify.targets
com.fortify.sca.ScaMSBuildFrameworkTargets=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core/private-bin/sca/MSBuildPlugin/Framework/Fortify.targets
com.fortify.sca.SolverTimeout=15
com.fortify.sca.SqlLanguage=TSQL
com.fortify.sca.SuppressLowSeverity=true
com.fortify.sca.ThreadCount.NameTableLoading=1
com.fortify.sca.TypeInferenceFunctionTimeout=60
com.fortify.sca.TypeInferenceLanguages=javascript,typescript,python,ruby
com.fortify.sca.TypeInferencePhase0Timeout=300
com.fortify.sca.UnicodeInputFile=true
com.fortify.sca.UniversalBlacklist=.*yyparse.*
com.fortify.sca.alias.mode.csharp=fs
com.fortify.sca.alias.mode.javascript=fi
com.fortify.sca.alias.mode.scala=fi
com.fortify.sca.alias.mode.swift=fs
com.fortify.sca.alias.mode.typescript=fi
com.fortify.sca.alias.mode.vb=fs
com.fortify.sca.analyzer.controlflow.EnableLivenessOptimization=false
com.fortify.sca.analyzer.controlflow.EnableMachineFiltering=false
com.fortify.sca.analyzer.controlflow.EnableRefRuleOptimization=false
com.fortify.sca.analyzer.controlflow.EnableTimeOut=true
com.fortify.sca.compilers.ant=com.fortify.sca.util.compilers.AntAdapter
com.fortify.sca.compilers.ar=com.fortify.sca.util.compilers.ArUtil
com.fortify.sca.compilers.armcc=com.fortify.sca.util.compilers.ArmCcCompiler
com.fortify.sca.compilers.armcpp=com.fortify.sca.util.compilers.ArmCppCompiler
com.fortify.sca.compilers.c++=com.fortify.sca.util.compilers.GppCompiler
com.fortify.sca.compilers.c89=com.fortify.sca.util.compilers.C89Compiler
com.fortify.sca.compilers.cc=com.fortify.sca.util.compilers.GccCompiler
com.fortify.sca.compilers.cl=com.fortify.sca.util.compilers.MicrosoftCompiler
com.fortify.sca.compilers.clearmake=com.fortify.sca.util.compilers.TouchlessCompiler
com.fortify.sca.compilers.devenv=com.fortify.sca.util.compilers.DevenvAdapter
com.fortify.sca.compilers.dotnet=com.fortify.sca.util.compilers.DotnetAdapter
com.fortify.sca.compilers.fortify=com.fortify.sca.util.compilers.FortifyCompiler
com.fortify.sca.compilers.g++=com.fortify.sca.util.compilers.GppCompiler
com.fortify.sca.compilers.g++-*=com.fortify.sca.util.compilers.GppCompiler
com.fortify.sca.compilers.g++2*=com.fortify.sca.util.compilers.GppCompiler
com.fortify.sca.compilers.g++3*=com.fortify.sca.util.compilers.GppCompiler
com.fortify.sca.compilers.g++4*=com.fortify.sca.util.compilers.GppCompiler
com.fortify.sca.compilers.gcc=com.fortify.sca.util.compilers.GccCompiler
com.fortify.sca.compilers.gcc-*=com.fortify.sca.util.compilers.GccCompiler
com.fortify.sca.compilers.gcc2*=com.fortify.sca.util.compilers.GccCompiler
com.fortify.sca.compilers.gcc3*=com.fortify.sca.util.compilers.GccCompiler
com.fortify.sca.compilers.gcc4*=com.fortify.sca.util.compilers.GccCompiler
com.fortify.sca.compilers.gmake=com.fortify.sca.util.compilers.TouchlessCompiler
com.fortify.sca.compilers.gradle=com.fortify.sca.util.compilers.GradleAdapter
com.fortify.sca.compilers.gradlew=com.fortify.sca.util.compilers.GradleAdapter
com.fortify.sca.compilers.icc=com.fortify.sca.util.compilers.IntelCompiler
com.fortify.sca.compilers.icl=com.fortify.sca.util.compilers.MicrosoftCompiler
com.fortify.sca.compilers.icpc=com.fortify.sca.util.compilers.IntelCompiler
com.fortify.sca.compilers.jam=com.fortify.sca.util.compilers.TouchlessCompiler
com.fortify.sca.compilers.javac=com.fortify.sca.util.compilers.JavacCompiler
com.fortify.sca.compilers.ld=com.fortify.sca.util.compilers.LdCompiler
com.fortify.sca.compilers.link=com.fortify.sca.util.compilers.MicrosoftLinker
com.fortify.sca.compilers.make=com.fortify.sca.util.compilers.TouchlessCompiler
com.fortify.sca.compilers.msbuild=com.fortify.sca.util.compilers.MSBuildAdapter
com.fortify.sca.compilers.msdev=com.fortify.sca.util.compilers.MSDevAdapter
com.fortify.sca.compilers.mvn=com.fortify.sca.util.compilers.MavenAdapter
com.fortify.sca.compilers.nmake=com.fortify.sca.util.compilers.TouchlessCompiler
com.fortify.sca.compilers.scalac=com.fortify.sca.util.compilers.ScalacCompiler
com.fortify.sca.compilers.tcc=com.fortify.sca.util.compilers.ArmCcCompiler
com.fortify.sca.compilers.tcpp=com.fortify.sca.util.compilers.ArmCppCompiler
com.fortify.sca.compilers.touchless=com.fortify.sca.util.compilers.FortifyCompiler
com.fortify.sca.compilers.xilink=com.fortify.sca.util.compilers.MicrosoftLinker
com.fortify.sca.dart.Enable=true
com.fortify.sca.dart.SDK=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core/dart/sdk/dart-sdk/
com.fortify.sca.dart.Translator=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core/private-bin/sca/dart/dart2nst.exe
com.fortify.sca.env.exesearchpath=C:\Program Files\Fortify\Fortify_Apps_and_Tools_23.1.0\bin;C:/Program
Files/Fortify/Fortify_Apps_and_Tools_23.1.0/Core/private-bin/awb/../../../jre/bin/server;C:/Program
Files/Fortify/Fortify_Apps_and_Tools_23.1.0/Core/private-bin/awb/../../../jre/bin;C:\Program
Files\Fortify\Fortify_SCA_23.1.0\bin;C:\Program
Files\Fortify\Fortify_Apps_and_Tools_23.1.0\bin;C:\Python27\;C:\Python27\Scripts;C:\ProgramData\Boxstarter;C:\Program
Files (x86)\Common Files\Oracle\Java\javapath;C:\Program Files\HPE_Security\Fortify_SCA_and_Apps_17.20\bin;C:\Program
Files\Common Files\Autodesk Shared\AcDwgFilter\;C:\Program Files\Common Files\Autodesk Shared\Advance\;C:\Program
Files (x86)\Common Files\Autodesk Shared\;C:\Program Files\Microsoft
MPI\Bin\;c:\programs.x86\oracle\client_11.2.0\bin;d:\oracle\app\root\product\12.1.0\dbhome_1\bin;C:\ProgramData\Oracle\Java\
javapath;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerS
hell\v1.0\;C:\Program Files\Microsoft SQL Server\120\Tools\Binn\;C:\Program Files\Microsoft SQL Server\Client
SDK\ODBC\110\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\120\Tools\Binn\;C:\Program Files\Microsoft SQL
Server\120\DTS\Binn\;C:\Program Files (x86)\Microsoft SQL Server\120\Tools\Binn\ManagementStudio\;C:\Program Files
(x86)\Microsoft SQL Server\120\DTS\Binn\;C:\Program Files (x86)\Microsoft SQL Server\130\Tools\Binn\;C:\Program
Files\Microsoft SQL Server\130\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\130\DTS\Binn\;C:\Program
Files\Microsoft SQL Server\130\DTS\Binn\;C:\Program Files\Microsoft SQL Server\Client
SDK\ODBC\130\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\140\DTS\Binn\;C:\Program Files
(x86)\Skype\Phone\;C:\Program Files (x86)\Microsoft ASP.NET\ASP.NET Web Pages\v1.0\;C:\Program Files\Microsoft SQL
Server\110\Tools\Binn\;C:\Program Files\Common Files\Autodesk Shared\;C:\Program Files\Common Files\Graitec\;C:\Program
Files\Microsoft Windows Performance Toolkit\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files\Microsoft\Web Platform
Installer\;C:\ProgramData\chocolatey\bin;c:\programs.x64\BaseX\bin;C:\Program Files\Microsoft SQL Server Migration
Assistant for Access\bin\;C:\Program Files (x86)\HID Global\ActivClient\;C:\Program Files\HID
Global\ActivClient\;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\Windo
wsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files (x86)\Microsoft SQL
Server\110\DTS\Binn\;C:\Program Files (x86)\Microsoft SQL Server\150\DTS\Binn\;C:\Program Files
(x86)\Sennheiser\SenncomSDK\;C:\programs.x64\TortoiseGit\bin;C:\Program Files\Microsoft SQL Server\Client
SDK\ODBC\170\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\150\Tools\Binn\;C:\Program Files\Microsoft SQL
Server\150\Tools\Binn\;C:\Program Files\Microsoft SQL Server\150\DTS\Binn\;C:\Program Files\Azure Data
Studio\bin;C:\Program Files\dotnet\;C:\Program Files\Intel\WiFi\bin\;C:\Program Files\Common
Files\Intel\WirelessCommon\;C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common;C:\Program Files (x86)\Microsoft
SQL
Server\160\DTS\Binn\;C:\Users\jpyeron\AppData\Local\Microsoft\WindowsApps;C:\Users\jpyeron\AppData\Local\Microsoft\W
indowsApps;c:\programs.x86\nmap;C:\Program Files\Azure Data Studio\bin;C:\Program Files\Intel\WiFi\bin\;C:\Program
Files\Common Files\Intel\WirelessCommon\;;C:\Program
Files\Fortify\Fortify_Apps_and_Tools_23.1.0\bin\..\Core\lib;C:\Program Files\Fortify\Fortify_Apps_and_Tools_23.1.0\bin;
com.fortify.sca.fileextensions.ABAP=ABAP
com.fortify.sca.fileextensions.BSP=ABAP
com.fortify.sca.fileextensions.Config=XML
com.fortify.sca.fileextensions.Dockerfile=DOCKERFILE
com.fortify.sca.fileextensions.Master=ASPNET
com.fortify.sca.fileextensions.abap=ABAP
com.fortify.sca.fileextensions.appxmanifest=XML
com.fortify.sca.fileextensions.as=ACTIONSCRIPT
com.fortify.sca.fileextensions.asax=ASPNET
com.fortify.sca.fileextensions.ascx=ASPNET
com.fortify.sca.fileextensions.ashx=ASPNET
com.fortify.sca.fileextensions.asmx=ASPNET
com.fortify.sca.fileextensions.asp=ASP
com.fortify.sca.fileextensions.aspx=ASPNET
com.fortify.sca.fileextensions.axml=ASPNET
com.fortify.sca.fileextensions.baml=MSIL
com.fortify.sca.fileextensions.bas=VB6
com.fortify.sca.fileextensions.bsp=ABAP
com.fortify.sca.fileextensions.cbl=COBOL
com.fortify.sca.fileextensions.cfc=CFML
com.fortify.sca.fileextensions.cfm=CFML
com.fortify.sca.fileextensions.cfml=CFML
com.fortify.sca.fileextensions.cjs=TYPESCRIPT
com.fortify.sca.fileextensions.cls=VB6
com.fortify.sca.fileextensions.cob=COBOL
com.fortify.sca.fileextensions.conf=HOCON
com.fortify.sca.fileextensions.config=XML
com.fortify.sca.fileextensions.cpx=XML
com.fortify.sca.fileextensions.cs=CSHARP
com.fortify.sca.fileextensions.cscfg=XML
com.fortify.sca.fileextensions.csdef=XML
com.fortify.sca.fileextensions.cshtml=ASPNET
com.fortify.sca.fileextensions.ctl=VB6
com.fortify.sca.fileextensions.ctp=PHP
com.fortify.sca.fileextensions.cts=TYPESCRIPT
com.fortify.sca.fileextensions.dart=DART
com.fortify.sca.fileextensions.dll=MSIL
com.fortify.sca.fileextensions.dockerfile=DOCKERFILE
com.fortify.sca.fileextensions.erb=RUBY_ERB
com.fortify.sca.fileextensions.exe=MSIL
com.fortify.sca.fileextensions.faces=JSPX
com.fortify.sca.fileextensions.frm=VB6
com.fortify.sca.fileextensions.go=GO
com.fortify.sca.fileextensions.hcl=HCL
com.fortify.sca.fileextensions.htm=HTML
com.fortify.sca.fileextensions.html=HTML
com.fortify.sca.fileextensions.ini=JAVA_PROPERTIES
com.fortify.sca.fileextensions.java=JAVA
com.fortify.sca.fileextensions.js=TYPESCRIPT
com.fortify.sca.fileextensions.jsff=JSPX
com.fortify.sca.fileextensions.json=JSON
com.fortify.sca.fileextensions.jsp=JSP
com.fortify.sca.fileextensions.jspf=JSP
com.fortify.sca.fileextensions.jspx=JSPX
com.fortify.sca.fileextensions.jsx=TYPESCRIPT
com.fortify.sca.fileextensions.kt=KOTLIN
com.fortify.sca.fileextensions.kts=KOTLIN
com.fortify.sca.fileextensions.master=ASPNET
com.fortify.sca.fileextensions.mdl=MSIL
com.fortify.sca.fileextensions.mjs=TYPESCRIPT
com.fortify.sca.fileextensions.mod=MSIL
com.fortify.sca.fileextensions.mts=TYPESCRIPT
com.fortify.sca.fileextensions.mxml=MXML
com.fortify.sca.fileextensions.object=APEX_OBJECT
com.fortify.sca.fileextensions.page=VISUAL_FORCE
com.fortify.sca.fileextensions.php=PHP
com.fortify.sca.fileextensions.phtml=PHP
com.fortify.sca.fileextensions.pkb=PLSQL
com.fortify.sca.fileextensions.pkh=PLSQL
com.fortify.sca.fileextensions.pks=PLSQL
com.fortify.sca.fileextensions.plist=XML
com.fortify.sca.fileextensions.properties=JAVA_PROPERTIES
com.fortify.sca.fileextensions.py=PYTHON
com.fortify.sca.fileextensions.razor=ASPNET
com.fortify.sca.fileextensions.rb=RUBY
com.fortify.sca.fileextensions.scala=SCALA
com.fortify.sca.fileextensions.settings=XML
com.fortify.sca.fileextensions.sql=SQL
com.fortify.sca.fileextensions.swift=SWIFT
com.fortify.sca.fileextensions.tag=JSP
com.fortify.sca.fileextensions.tagx=JSP
com.fortify.sca.fileextensions.tf=HCL
com.fortify.sca.fileextensions.tld=TLD
com.fortify.sca.fileextensions.trigger=APEX_TRIGGER
com.fortify.sca.fileextensions.ts=TYPESCRIPT
com.fortify.sca.fileextensions.tsx=TYPESCRIPT
com.fortify.sca.fileextensions.vb=VB
com.fortify.sca.fileextensions.vbhtml=ASPNET
com.fortify.sca.fileextensions.vbs=VBSCRIPT
com.fortify.sca.fileextensions.vbscript=VBSCRIPT
com.fortify.sca.fileextensions.vue=VUE
com.fortify.sca.fileextensions.wadcfg=XML
com.fortify.sca.fileextensions.wadcfgx=XML
com.fortify.sca.fileextensions.winmd=MSIL
com.fortify.sca.fileextensions.wsdd=XML
com.fortify.sca.fileextensions.wsdl=XML
com.fortify.sca.fileextensions.xaml=ASPNET
com.fortify.sca.fileextensions.xcfg=XML
com.fortify.sca.fileextensions.xhtml=JSPX
com.fortify.sca.fileextensions.xmi=XML
com.fortify.sca.fileextensions.xml=XML
com.fortify.sca.fileextensions.xsd=XML
com.fortify.sca.fileextensions.yaml=YAML
com.fortify.sca.fileextensions.yml=YAML
com.fortify.sca.jsp.UseNativeParser=true
com.fortify.sca.lim.RequireTrustedSSLCert=true
com.fortify.sca.parser.python.ignore.module.1=test.badsyntax_future3
com.fortify.sca.parser.python.ignore.module.2=test.badsyntax_future4
com.fortify.sca.parser.python.ignore.module.3=test.badsyntax_future5
com.fortify.sca.parser.python.ignore.module.4=test.badsyntax_future6
com.fortify.sca.parser.python.ignore.module.5=test.badsyntax_future7
com.fortify.sca.parser.python.ignore.module.6=test.badsyntax_future8
com.fortify.sca.parser.python.ignore.module.7=test.badsyntax_future9
com.fortify.sca.parser.python.ignore.module.8=test.badsyntax_nocaret
com.fortify.sca.skip.libraries.AngularJS=angular.js,angular.min.js,angular-animate.js,angular-aria.js,angular_1_router.js,angular-cookies.js,angular-message-format.js,angular-messages.js,angular-mocks.js,angular-parse-ext.js,angular-resource.js,angular-route.js,angular-sanitize.js,angular-touch.js
com.fortify.sca.skip.libraries.ES6=es6-shim.min.js,system-polyfills.js,shims_for_IE.js
com.fortify.sca.skip.libraries.jQuery=jquery.js,jquery.min.js,jquery-migrate.js,jquery-migrate.min.js,jquery-ui.js,jquery-ui.min.js,jquery.mobile.js,jquery.mobile.min.js,jquery.color.js,jquery.color.min.js,jquery.color.svg-names.js,jquery.color.svg-names.min.js,jquery.color.plus-names.js,jquery.color.plus-names.min.js,jquery.tools.min.js
com.fortify.sca.skip.libraries.javascript=bootstrap.js,bootstrap.min.js,typescript.js,typescriptServices.js
com.fortify.sca.skip.libraries.typescript=typescript.d.ts,typescriptServices.d.ts
com.fortify.search.defaultSyntaxVer=2
com.sun.management.jmxremote=true
dotnet.install.dir=C:\Windows\Microsoft.NET\Framework64\
dotnet.sdk.v11.install.dir=
dotnet.sdk.v20.install.dir=
dotnet.sdk.v3x.install.dir=C:\Program Files\Microsoft SDKs\Windows\v6.0A\
dotnet.v30.referenceAssemblies=C:\Program Files\Reference Assemblies\Microsoft\Framework\v3.0\
dotnet.v35.referenceAssemblies=C:\Program Files\Reference Assemblies\Microsoft\Framework\v3.5\
file.encoding=Cp1252
file.separator=\
ide.hide.excluded.files=false
idea.ignore.disabled.plugins=true
idea.io.use.nio2=true
idea.plugins.compatible.build=999.SNAPSHOT
java.awt.graphicsenv=sun.awt.Win32GraphicsEnvironment
java.awt.headless=true
java.awt.printerjob=sun.awt.windows.WPrinterJob
java.class.path=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core\lib\exe\sca-exe.jar
java.class.version=55.0
java.home=C:\Program Files\Fortify\Fortify_SCA_23.1.0\jre
java.io.tmpdir=C:\Users\jpyeron\AppData\Local\Temp\
java.library.path=C:\Program Files\Fortify\Fortify_SCA_23.1.0\bin;C:\WINDOWS\Sun\Java\bin;C:\WINDOWS\system32;C:\WINDOWS;C:/Program Files/Fortify/Fortify_Apps_and_Tools_23.1.0/Core/private-bin/awb/../../../jre/bin/server;C:/Program Files/Fortify/Fortify_Apps_and_Tools_23.1.0/Core/private-bin/awb/../../../jre/bin;C:\Program Files\Fortify\Fortify_SCA_23.1.0\bin;C:\Program Files\Fortify\Fortify_Apps_and_Tools_23.1.0\bin;C:\Python27\;C:\Python27\Scripts;C:\ProgramData\Boxstarter;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Program Files\HPE_Security\Fortify_SCA_and_Apps_17.20\bin;C:\Program Files\Common Files\Autodesk Shared\AcDwgFilter\;C:\Program Files\Common Files\Autodesk Shared\Advance\;C:\Program Files (x86)\Common Files\Autodesk Shared\;C:\Program Files\Microsoft MPI\Bin\;c:\programs.x86\oracle\client_11.2.0\bin;d:\oracle\app\root\product\12.1.0\dbhome_1\bin;C:\ProgramData\Oracle\Java\javapath;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\Program Files\Microsoft SQL Server\120\Tools\Binn\;C:\Program Files\Microsoft SQL Server\Client
```

```
SDK\ODBC\110\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\120\Tools\Binn\;C:\Program Files\Microsoft SQL
Server\120\DTS\Binn\;C:\Program Files (x86)\Microsoft SQL Server\120\Tools\Binn\ManagementStudio\;C:\Program Files
(x86)\Microsoft SQL Server\120\DTS\Binn\;C:\Program Files (x86)\Microsoft SQL Server\130\Tools\Binn\;C:\Program
Files\Microsoft SQL Server\130\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\130\DTS\Binn\;C:\Program
Files\Microsoft SQL Server\130\DTS\Binn\;C:\Program Files\Microsoft SQL Server\Client
SDK\ODBC\130\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\140\DTS\Binn\;C:\Program Files
(x86)\Skype\Phone\;C:\Program Files (x86)\Microsoft ASP.NET\ASP.NET Web Pages\v1.0\;C:\Program Files\Microsoft SQL
Server\110\Tools\Binn\;C:\Program Files\Common Files\Autodesk Shared\;C:\Program Files\Common Files\Graitec\;C:\Program
Files\Microsoft Windows Performance Toolkit\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files\Microsoft\Web Platform
Installer\;C:\ProgramData\chocolatey\bin;c:\programs.x64\BaseX\bin;C:\Program Files\Microsoft SQL Server Migration
Assistant for Access\bin\;C:\Program Files (x86)\HID Global\ActivClient\;C:\Program Files\HID
Global\ActivClient\;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\Windo
wsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files (x86)\Microsoft SQL
Server\110\DTS\Binn\;C:\Program Files (x86)\Microsoft SQL Server\150\DTS\Binn\;C:\Program Files
(x86)\Sennheiser\SenncomSDK\;C:\programs.x64\TortoiseGit\bin;C:\Program Files\Microsoft SQL Server\Client
SDK\ODBC\170\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\150\Tools\Binn\;C:\Program Files\Microsoft SQL
Server\150\Tools\Binn\;C:\Program Files\Microsoft SQL Server\150\DTS\Binn\;C:\Program Files\Azure Data
Studio\bin;C:\Program Files\dotnet\;C:\Program Files\Intel\WiFi\bin\;C:\Program Files\Common
Files\Intel\WirelessCommon\;C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common;C:\Program Files (x86)\Microsoft
SQL
Server\160\DTS\Binn\;C:\Users\jpyeron\AppData\Local\Microsoft\WindowsApps;C:\Users\jpyeron\AppData\Local\Microsoft\W
indowsApps;c:\programs.x86\nmap;C:\Program Files\Azure Data Studio\bin;C:\Program Files\Intel\WiFi\bin\;C:\Program
Files\Common Files\Intel\WirelessCommon\;;C:\Program
Files\Fortify\Fortify_Apps_and_Tools_23.1.0\bin\..\Core\lib;C:\Program Files\Fortify\Fortify_Apps_and_Tools_23.1.0\bin;;.
java.rmi.server.randomIDs=true
java.runtime.name=OpenJDK Runtime Environment
java.runtime.version=11.0.18+10-LTS
java.specification.name=Java Platform API Specification
java.specification.vendor=Oracle Corporation
java.specification.version=11
java.vendor=Azul Systems, Inc.
java.vendor.url=http://www.azul.com/
java.vendor.url.bug=http://www.azul.com/support/
java.vendor.version=Zulu11.62+17-CA
java.version=11.0.18
java.version.date=2023-01-17
java.vm.info=mixed mode
java.vm.name=OpenJDK 64-Bit Server VM
java.vm.specification.name=Java Virtual Machine Specification
java.vm.specification.vendor=Oracle Corporation
java.vm.specification.version=11
java.vm.vendor=Azul Systems, Inc.
java.vm.version=11.0.18+10-LTS
jdk.debug=release
jdk.vendor.version=Zulu11.62+17-CA
line.separator=

log4j.configurationFile=C:\Program Files\Fortify\Fortify_SCA_23.1.0\Core\config\log4j2.xml
log4j.isThreadContextMapInheritable=true
max.file.path.length=255
os.arch=amd64
os.name=Windows 10
os.version=10.0
path.separator=;
project.structure.add.tools.jar.to.new.jdk=false
psi.incremental.reparse.depth.limit=1000
psi.track.invalidation=true
stderr.isatty=false
stdout.isatty=false
sun.arch.data.model=64
sun.boot.library.path=C:\Program Files\Fortify\Fortify_SCA_23.1.0\jre\bin
sun.cpu.endian=little
sun.cpu.isalist=amd64
sun.desktop=windows
sun.io.unicode.encoding=UnicodeLittle
sun.java.command=sourceanalyzer -Djava.awt.headless=true -Dcom.sun.management.jmxremote=true -XX:+UseParallelGC -
XX:SoftRefLRUPolicyMSPerMB=3000 --illegal-access=permit --add-
exports=jdk.management/com.sun.management.internal=ALL-UNNAMED --add-
exports=jdk.scripting.nashorn/jdk.nashorn.internal.runtime=ALL-UNNAMED --add-exports=java.base/jdk.internal.misc=ALL-
UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-
opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/sun.security.jca=ALL-UNNAMED --add-
opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-
opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.util.regex=ALL-UNNAMED --add-
opens=java.base/java.net=ALL-UNNAMED --add-opens=java.base/javax.crypto=ALL-UNNAMED --add-
opens=java.management/sun.management=ALL-UNNAMED -Dwin32.LocalAppdata=C:\Users\jpyeron\AppData\Local -
Ddotnet.install.dir=C:\Windows\Microsoft.NET\Framework64\ -Ddotnet.sdk.v11.install.dir= -Ddotnet.sdk.v20.install.dir= -
Ddotnet.sdk.v3x.install.dir=C:\Program Files\Microsoft SDKs\Windows\v6.0A\ -Ddotnet.v30.referenceAssemblies=C:\Program
Files\Reference Assemblies\Microsoft\Framework\v3.0\ -Ddotnet.v35.referenceAssemblies=C:\Program Files\Reference
Assemblies\Microsoft\Framework\v3.5\ -Dvs.110.dotnet.clr.version=v4.0.30319 -Dvs.140.dotnet.clr.version=v4.0.30319 -
Dcom.fortify.sca.env.exesearchpath=C:\Program Files\Fortify\Fortify_Apps_and_Tools_23.1.0\bin;C:/Program
Files/Fortify/Fortify_Apps_and_Tools_23.1.0/Core/private-bin/awb/../../../jre/bin/server;C:/Program
Files/Fortify/Fortify_Apps_and_Tools_23.1.0/Core/private-bin/awb/../../../jre/bin;C:\Program
Files\Fortify\Fortify_SCA_23.1.0\bin;C:\Program
Files\Fortify\Fortify_Apps_and_Tools_23.1.0\bin;C:\Python27\;C:\Python27\Scripts;C:\ProgramData\Boxstarter;C:\Program
Files (x86)\Common Files\Oracle\Java\javapath;C:\Program Files\HPE_Security\Fortify_SCA_and_Apps_17.20\bin;C:\Program
Files\Common Files\Autodesk Shared\AcDwgFilter\;C:\Program Files\Common Files\Autodesk Shared\Advance\;C:\Program
Files (x86)\Common Files\Autodesk Shared\;C:\Program Files\Microsoft
MPI\Bin\;c:\programs.x86\oracle\client_11.2.0\bin;d:\oracle\app\root\product\12.1.0\dbhome_1\bin;C:\ProgramData\Oracle\Java\
javapath;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerS
hell\v1.0\;C:\Program Files\Microsoft SQL Server\120\Tools\Binn\;C:\Program Files\Microsoft SQL Server\Client
SDK\ODBC\110\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\120\Tools\Binn\;C:\Program Files\Microsoft SQL
Server\120\DTS\Binn\;C:\Program Files (x86)\Microsoft SQL Server\120\Tools\Binn\ManagementStudio\;C:\Program Files
(x86)\Microsoft SQL Server\120\DTS\Binn\;C:\Program Files (x86)\Microsoft SQL Server\130\Tools\Binn\;C:\Program
Files\Microsoft SQL Server\130\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\130\DTS\Binn\;C:\Program
Files\Microsoft SQL Server\130\DTS\Binn\;C:\Program Files\Microsoft SQL Server\Client
SDK\ODBC\130\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\140\DTS\Binn\;C:\Program Files
(x86)\Skype\Phone\;C:\Program Files (x86)\Microsoft ASP.NET\ASP.NET Web Pages\v1.0\;C:\Program Files\Microsoft SQL
Server\110\Tools\Binn\;C:\Program Files\Common Files\Autodesk Shared\;C:\Program Files\Common Files\Graitec\;C:\Program
Files\Microsoft Windows Performance Toolkit\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files\Microsoft\Web Platform
Installer\;C:\ProgramData\chocolatey\bin;c:\programs.x64\BaseX\bin;C:\Program Files\Microsoft SQL Server Migration
Assistant for Access\bin\;C:\Program Files (x86)\HID Global\ActivClient\;C:\Program Files\HID
Global\ActivClient\;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\Windo
```

```
wsPowerShell\v1.0\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files (x86)\Microsoft SQL
Server\110\DTS\Binn\;C:\Program Files (x86)\Microsoft SQL Server\150\DTS\Binn\;C:\Program Files
(x86)\Sennheiser\SenncomSDK\;C:\programs.x64\TortoiseGit\bin;C:\Program Files\Microsoft SQL Server\Client
SDK\ODBC\170\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\150\Tools\Binn\;C:\Program Files\Microsoft SQL
Server\150\Tools\Binn\;C:\Program Files\Microsoft SQL Server\150\DTS\Binn\;C:\Program Files\Azure Data
Studio\bin;C:\Program Files\dotnet\;C:\Program Files\Intel\WiFi\bin\;C:\Program Files\Common
Files\Intel\WirelessCommon\;C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common;C:\Program Files (x86)\Microsoft
SQL
Server\160\DTS\Binn\;C:\Users\jpyeron\AppData\Local\Microsoft\WindowsApps;C:\Users\jpyeron\AppData\Local\Microsoft\W
indowsApps;c:\programs.x86\nmap;C:\Program Files\Azure Data Studio\bin;C:\Program Files\Intel\WiFi\bin\;C:\Program
Files\Common Files\Intel\WirelessCommon\;;C:\Program
Files\Fortify\Fortify_Apps_and_Tools_23.1.0\bin\..\Core\lib;C:\Program Files\Fortify\Fortify_Apps_and_Tools_23.1.0\bin; -
Dcom.fortify.sca.ProjectRoot=C:\Users\jpyeron\AppData\Local/Fortify -Dstdout.isatty=false -Dstderr.isatty=false -
Dcom.fortify.sca.PID=33416 -Xmx35010M -Dcom.fortify.TotalPhysicalMemory=68388900864 -Xss16M -
Dcom.fortify.sca.JVMArgs=-XX:+UseParallelGC -XX:SoftRefLRUPolicyMSPerMB=3000 --illegal-access=permit --add-
exports=jdk.management/com.sun.management.internal=ALL-UNNAMED --add-
exports=jdk.scripting.nashorn/jdk.nashorn.internal.runtime=ALL-UNNAMED --add-exports=java.base/jdk.internal.misc=ALL-
UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-
opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/sun.security.jca=ALL-UNNAMED --add-
opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-
opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.util.regex=ALL-UNNAMED --add-
opens=java.base/java.net=ALL-UNNAMED --add-opens=java.base/javax.crypto=ALL-UNNAMED --add-
opens=java.management/sun.management=ALL-UNNAMED -Xmx35010M -Xss16M -Djava.class.path=C:\Program
Files\Fortify\Fortify_SCA_23.1.0\Core\lib\exe\sca-exe.jar -scan -pid-file
C:\Users\jpyeron\AppData\Local\Temp\PID6845777546103778301.tmp @C:\Users\jpyeron\AppData\Local\Fortify\AWB-
23.1.0\modheader\modheaderScan.txt
sun.jnu.encoding=Cp1252
sun.management.compiler=HotSpot 64-Bit Tiered Compilers
sun.os.patch.level=
user.country=US
user.dir=C:\Program Files\Fortify\Fortify_Apps_and_Tools_23.1.0\bin
user.home=C:\Users\jpyeron
user.language=en
user.name=jpyeron
user.script=
user.timezone=America/New_York
user.variant=
vs.110.dotnet.clr.version=v4.0.30319
vs.140.dotnet.clr.version=v4.0.30319
win32.LocalAppdata=C:\Users\jpyeron\AppData\Local
```

Commandline Arguments

```
-scan
-pid-file
C:\Users\jpyeron\AppData\Local\Temp\PID6845777546103778301.tmp
-b
modheader
-machine-output
-format
fpr
```

-f
C:\Users\jpyeron\AppData\Local/Fortify\AWB-23.1.0\modheader\modheader.fpr

Warnings

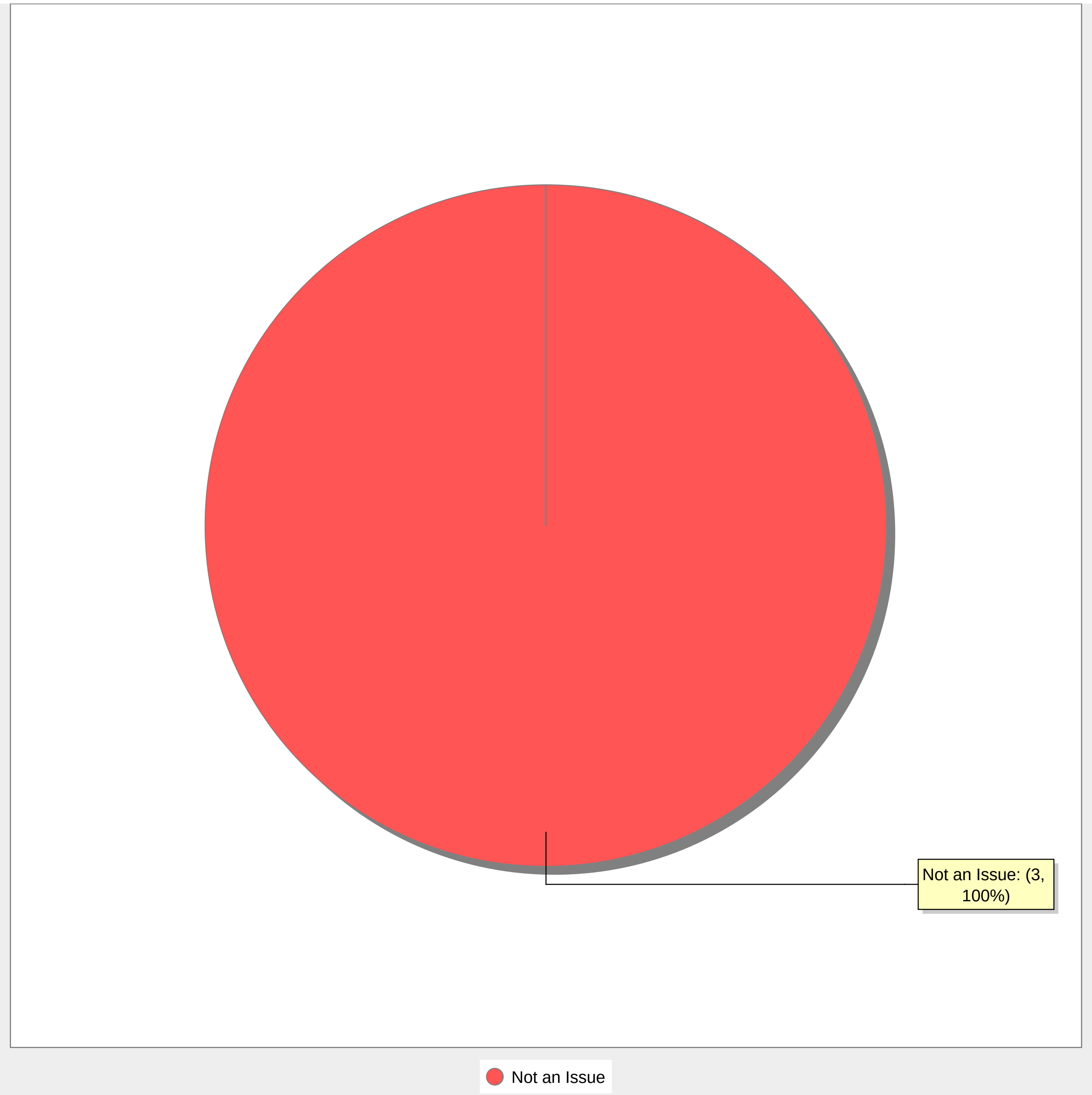
No warnings occurred during analysis

Issue Count by Category

Issues by Category	
Cross-Site Scripting: DOM	1
Insecure Randomness	1
Open Redirect	1

Issue Breakdown by Analysis

Issues by Analysis


Not an Issue: (3, 100%)
Not an Issue