Pulse Connect Secure

How-to Guide for WSAM access to debug rewriter issues.
Configuring test WSAM account for debugging by engineering and support.

The Pulse Secure support and engineering teams need to be able to replicate rewrite issues in-house in order to identify the problem and provide a fix. When the problematic site is configured for access via WSAM on the PCS device, the Pulse Secure team accesses the test URL provided and launches WSAM. We can then use a proxy to route all traffic on the machine where WSAM tunnel is running to customer’s environment and configure a web proxy policy on an internal PCS device to send web traffic via that proxy. With this setup, we will have the application configured as a web bookmark in our internal PCS device and accessed via rewrite. We can then replicate and debug the issue on our internal PCS devices.

Why we need WSAM test access for debugging Rewriter issues:

• 90% of customer issues are client side, where we need to debug the client side JavaScript or other web technologies to see why the issue is happening. In such cases, mere logs are of not much use for us as code is obfuscated and not straight forward.

• We need to put debugging breakpoints in the customer application code to figure out the issue. For this we need a proper replication set up. Also when client uses ActiveX, dsrecord/session recording log cannot be used.

• Most of the time the problematic requests are generated using ajax calls and hence cannot be tracked using logs. We need to get to the proper ajax call using direct case replication and then debug the rewriter case.

• Also debugging the rewriter issue in customer environment is highly inefficient, as the rewriter client side code [ie.js/oth.js] is completely scrambled [function parameters are replaced with random text].

• Debugging in the customer environment, on customer desktop using remote sessions is inefficient as the customer environment won’t have required debugging tools and sometimes the language used by the customer browsers pose a great challenge to the developers.

• If test WSAM access for debugging cannot be provided, we may need to do supervised debugging session if logs are not sufficient to debug and fix issue. Collaborating engineer & customer availability for remote session results in delay of issue resolution due to time zone difference and also multiple rounds of such sessions may be needed as one session may not be sufficient.

Advantages of providing test WSAM access for debugging:

• Provides engineering with a 24/7 debugging environment leading to faster resolution.

• Engineering/support can test on higher PCS OS versions and check if issue is there on newer versions.

• Pass-through proxy can be tested on our end to check if it resolves the issue and can be suggested as an alternative clientless access method.
Configuring WSAM access for debugging:

Step 1:
Configure a test Role for example Pulsesecuretest and enable web and WSAM for the role.

Step 2:
Configure a test user Realm, assign the test user the test role and create the system local user.
Step 3:

Create a sign-in policy that maps to the user Realm.

<table>
<thead>
<tr>
<th>User URLs</th>
<th>Sign-In Page</th>
<th>Authentication Realm(s)</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>* /dev</td>
<td>Default Sign-In Page</td>
<td>Users</td>
<td>✓</td>
</tr>
<tr>
<td>* /debug</td>
<td>Default Sign-In Page</td>
<td>Users</td>
<td>✓</td>
</tr>
<tr>
<td>* /admin</td>
<td>Default Sign-In Page</td>
<td>Users</td>
<td>✓</td>
</tr>
<tr>
<td>* /support</td>
<td>Default Sign-In Page</td>
<td>Pulsesupport</td>
<td>✓</td>
</tr>
</tbody>
</table>

Step 4:

Create a web resource profile as below and click Show All auto policy types.

Navigate to Autopolicy: Rewriting options and select option below and Save and assign the web resource profile to the test role.
Note: There may be multiple server URL’s for the web application for example a redirect to another URL, we need to add all those server URL and ports under WSAM destinations.

Step 5:

Provide support the test login details for example:

https://PCSURL/pulsedebu

Test username: password:

Backend web application URL:

Application test username:

Application test password:

Note: If the backend web application contains confidential data/information and supervised debugging session is required, have the test access ready. We will require someone from the customer’s/partner’s end to be available for a remote debugging session with engineering and support team to debug the issue on call as we will need someone to provide the credentials to access the application on the developer’s/support team’s machine on which we are debugging the issue.
Related articles/documentation:

For best practices on CIE Best Practices, refer

For Troubleshooting Rewrite related issues and log collection, refer

For Pass-through proxy configuration which is an alternative to rewrite, refer
https://www.pulsesecure.net/download/techpubs/current/434/pulse-connectsecure/pcs/8.2rx/How_To_PTP.pdf