Layer7 SiteMinder

How SiteMinder Avoids Impact of the Default Behavior of Google Chrome 80 for SameSite Cookie Attribute

CA Technologies, a Broadcom Company, Documentation.

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Contents

How SiteMinder Avoids Impact of Default Behavior of Google Chrome 80 for SameSite Cookie Attribute

What is SameSite Cookie Attribute in Google Chrome 80

When a request is sent from a web browser to website, the web browser verifies whether it has stored any cookies for the website. If the browser finds a cookie, it validates the cookie properties and flags to understand whether to send the cookie in the transaction to the website.

While this behavior enables a better user experience, it creates a security risk that can lead to CSRF vulnerabilities. To help mitigate this risk, Google Chrome is changing how it enforces its default behavior based on the SameSite cookie attribute. This change is effective from Chrome 80.

Google Chrome will start enforcing the SameSite cookie attribute from the upcoming release of Chrome 80 to govern its default cookie management behavior. This cookie attribute determines whether browsers will send stored cookies to cross-site websites. For detailed information about the upcoming changes in Google Chrome 80, see Google documentation.

The SameSite cookie attribute is in the following format:

Set-Cookie: CookieName=CookieValue; SameSite=Strict|Lax|None;

The possible values are:

Strict

Specifies that the cookie will be sent only in requests that are initiated from the same site websites but not in requests that are initiated by cross-site websites.

Lax

Specifies that the cookie is sent in requests that are initiated from the same site websites, and in idempotent requests such as GET to cross-site websites. The cookie is not sent in non-idempotent requests such as POST to cross-site websites. This is the default behavior of Chrome 80.

None

Specifies that the existing behavior of sending stored cookies in requests initiated from same-site or cross-site websites continues.

This document covers information about how SiteMinder can be configured to avoid the impact of Chrome 80 in regard to the SameSite cookie attribute.

How Google Chrome 80 Default Behavior Affects SiteMinder

A core functionality of SiteMinder is to manage cross-website access. As the SameSite cookie attribute is designed to improve the security of cross-site cookie usage, the new default behavior of Google Chrome 80 will affect your SiteMinder environment if all the following conditions are met in your environment:

- Transactions are initiated from cross-sites. For example, a transaction is initiated by accessing a link in an application that is deployed on another domain
- Accessing a link initiates a request with non-idempotent method like POST

The following sections in this document explain in detail how SameSite cookie attribute affects SiteMinder.

Effect of Google Chrome 80 Default Behavior on Non-Federation Functionality of SiteMinder in Cross-Sites

This section describes whether Google Chrome 80 default behavior will affect the non-federation functionality of SiteMinder, that is, use cases that *do not involve* SAML, WS-FED, or OpenID Connect.

List of Use Cases that Will Fail

The following use cases will fail functionally with the default behavior of Chrome 80:

- Cookie provider flow for any POST request to an application
- Custom cookies that are generated using responses when a POST request is initiated from a cross-site
- Auditing of Anonymous authentication scheme may result in inconsistent results
- Basic authentication scheme when a non-idempotent request like a POST request is initiated from a cross-site

List of Use Cases that Will Succeed with a Change in User Experience

The following use cases will succeed with a change in user experience (for example, multiple user re-authentications may be required) with the default behavior of Chrome 80:

- SSO between applications when a SMSESSION cookie exists and a POST request is initiated from a cross-site
- SSO between applications when a session scheme mini cookie exists and a POST request is initiated from a cross-site

Effect of Google Chrome 80 Default Behavior on Federation Functionality of SiteMinder

This section describes whether Google Chrome 80 default behavior will affect the federation functionality of SiteMinder that *involves* SAML, WS-FED, or OpenID Connect.

List of Use Cases that Will Fail

The following federation use cases will fail functionally with the default behavior of Chrome 80:

- SP-initiated federation SSO transactions with the RelayState query parameter in SAML 2.0
- SAML 2.0 SLO with HTTP-POST binding
- SiteMinder acts as IdP in SAML 2.0 and Authentication Mode is set to Credential Selector (CHS)
- ForceAuthn in SAML 2.0 when SiteMinder acts as IdP and Authentication Requesting Binding is HTTP-POST
- Logout at IdP when SiteMinder acts as IdP and SP in WS-FED SLO
- AJAX requests for Single Page Applications in OpenID Connect
- Re-authentication at Authorization Endpoint (prompt=login) when SiteMinder acts as OpenID Connect Provider

Note that there are no federation use cases that will succeed with a change in user experience.

How to Configure SiteMinder to Manage the Change in the Default Behavior of Google Chrome 80

To support the SameSite cookie attribute of Google Chrome 80 and to ensure that you continue to have a smooth user experience, SiteMinder now provides two new Agent Configuration Object parameters, SameSite and getcpcookie (for cookie provider Agents). The ACO parameters let you control the default behavior of applications for SameSite cookie attribute in cross-site requests.

The following sections provide information on the solution.

Analyze the Impact of Google Chrome 80 default behavior

Assess how your SiteMinder environment is affected by Google Chrome 80 default behavior.

Examples:

- What exactly are the changes in Google Chrome 80? Read Google documentation for detailed information about the upcoming changes in Google Chrome 80.
- Is my environment implementing any of the use cases listed in the *Effect of Google Chrome 80 Default Behavior on Non-Federation Functionality of SiteMinder* section?
- Does my environment include cross-site requests using the POST method?
- Do the Agents in my environment use cookie provider functionality where Agents and cookie provider Agents are in different domains?
- Is federation implemented in my environment for any of the use cases listed in the *Effect of Google Chrome 80 Default Behavior on Federation Functionality of SiteMinder* section?
- Does any of the SiteMinder components in your environment interact with users who use Google Chrome 80?

If the analysis determines that your SiteMinder environment will be affected, you can further check if your organization has the ability to control Chrome Policies. If you can control the policies, you can choose to set Chrome 80 to have the pre-Chrome 80 behavior and avoid the impact of the change in Chrome. This action will provide you additional time for considering whether you want to deploy the SiteMinder solution for SameSite cookie attribute or to continue having Chrome function as it did prior to Chrome 80.

To manually control the default behavior of Google Chrome 80, Google Chrome provides two flags, **SameSite by default cookies** and **Cookies without SameSite must be secure**. For detailed information about configuring Chrome 80 with pre-Chrome 80 behavior, see Google Chrome documentation.

Deploy the SiteMinder Solution

If you will be affected by Chrome 80 default behavior, verify whether the affected use cases use any of the following components in their transactions:

- Access Gateway
- Web Agents
- Web Agent Option Pack

- Agent for SharePoint
- Web Services Security (WSS) Agent
- Application Server Agent (ASA) for Oracle WebLogic Server

If you are using any of the listed components, deploy the binaries that are provided as part of the SiteMinder solution to address the Chrome 80 changes.

Once you deploy the binaries, SiteMinder provides the following two new ACO parameters:

SameSite

Controls the behavior of SameSite cookie attribute. For more information about this ACO parameter, see the *Configure SameSite ACO Parameter in SiteMinder Administrative UI* section.

getcpcookie

Controls how requests to cookie provider Agents work with SameSite cookie attribute of Chrome 80. For more information about this ACO parameter, see *Configure getcpcookie ACO Parameter in SiteMinder Administrative UI* section.

Follow these steps:

1. Download the provided binaries.
2. Stop the components you want to upgrade.
3. Copy the required binaries into the specified location and start the components:

Important! Take a backup of all the existing binaries you are going to replace.

Component/Flow	Binary Name	Installation Location
Access Gateway on Windows	■ fedfws_dash o.jar ■ fedutil.jar	/secure-proxy/Tomcat/webapps/affwebservi ces/WEB-INF/lib
	■ HTTPPlugin. dll ■ SAMLDataPL ugin.dll ■ SPS60Agent .dll	/secure-proxy/agentframework/bin
	■ ProxyCore.j ar ■ Proxyrt.jar	/secure-proxy/Tomcat/lib

Component/Flow	Binary Name	Installation Location
Access Gateway on UNIX	■ fedfws_dash o.jar ■ fedutil.jar	/secure-proxy/Tomcat/webapps/affwebservi ces/WEB-INF/lib
	■ libHttpPlugi n.so ■ libSAMLDat aPlugin.so ■ libSPS60Ag ent.so	/secure-proxy/agentframework/bin
	■ proxyCore.j ar ■ proxyrt.jar	/secure-proxy/Tomcat/lib
Federation flow (12.52.x or lower) for Access Gateway and Web Agent Option Pack	■ fedfws_dash o.jar ■ fedutil.jar	/affwebservices/WEB-INF/lib
Federation flow (12.6 or higher) for Access Gateway	■ fedutil.jar ■ fedfws_obfs .jar	*AccessGateway_installation_path*/s ecure-proxy/Tomcat/webapps/affwebservi ces/WEB-INF/lib
Web Agent on Windows 64-bit	■ HTTPPlugin. dll ■ SAMLDataPL ugin.dll	■ 64-bit: webagent/win64/bin ■ 32-bit: webagent/win32/bin
Web Agent on Windows 32-bit	■ HTTPPlugin. dll ■ SAMLDataPL ugin.dll	webagent/bin

Component/Flow	Binary Name	Installation Location
Web Agent on UNIX	■ libHttpPlugin.so ■ libSAMLDataPlugin.so	webagent/bin
Agent for SharePoint on Windows	■ fedfws_dasho.jar ■ fedutil.jar	/Agent-for-SharePoint/Tomcat/webapps/affwebservices/WEB-INF/lib
	■ HTTPPlugin.dll ■ SPConfigModule.dll ■ SPPlugin.dll ■ SPS60Agent.dll	/Agent-for-SharePoint/agentframework/bin
	■ proxyCore.jar ■ proxyrt.jar	/Agent-for-SharePoint/Tomcat/lib
Agent for SharePoint on UNIX	■ fedfws_dasho.jar ■ fedutil.jar	/Agent-for-SharePoint/Tomcat/webapps/affwebservices/WEB-INF/lib
	■ libHttpPlugin.so ■ libSPConfigModule.so ■ libSPPlugin.so ■ libSPS60Agent.so	/Agent-for-SharePoint/agentframework/bin
	■ proxyCore.jar ■ proxyrt.jar	/Agent-for-SharePoint/Tomcat/lib

Component/Flow	Binary Name	Installation Location
Web Services Security Agent for WebLogic on Windows	■ soaagent-core.jar ■ soaagent-httpplugin.jar ■ soaagent-jaxwsplugin.jar	*WSS_Agent_Installation_Path*\lib
Web Services Security Agent for JBoss on Windows	■ soaagent-core.jar ■ soaagent-httpplugin.jar ■ soaagent-jaxwsplugin.jar	*JBoss_Server_Installation_Path*\modules\com\ca\siteminder\jbossagent\main
Web Services Security Agent for WebSphere on Windows	■ soaagent-core.jar ■ soaagent-httpplugin.jar ■ soaagent-jaxwsplugin.jar	*WebSphere_Server_Installation_Path*\ AppServer\lib\ext
Web Services Security Agent for Web Servers on Windows	■ HTTPPlug ■ SAMLDataPLugin.dll	webagent/bin
Application Server Agent (ASA) for WebLogic on Windows and UNIX	■ smsecurityproviders.jar	weblogic_installation_location\server\lib\mbeantypes

Configure SameSite ACO Parameter in SiteMinder Administrative UI

A new Agent Configuration Object (ACO), **SameSite**, is introduced to control the SameSite cookie attribute.

Follow these steps:

1. Open Administrative UI.
2. Navigate to Infrastructure, Agent, and Agent Configuration Objects.
3. Open the ACO for Agent.
4. Configure SameSite ACO parameter using one of the following values:

Note: The values are case-insensitive.

Blank

Specifies that if the ACO is not defined or not set to Strict, Lax, or None, the web browser settings are considered and no value is added to the SameSite cookie attribute. This is same as not adding the SameSite ACO parameter. This is the default value of the ACO parameter.

Strict

Specifies that the cookie will be sent only in requests that are initiated from the same site websites but not in requests that are initiated by cross- site websites. Use **Strict** to restrict all operations in a cross-site request.

Lax

Specifies that the cookie is sent in requests that are initiated from the same site websites, and in idempotent requests such as GET to cross-site websites. The cookie is not sent in non-idempotent requests such as POST to cross-site websites. Use **Lax** to allow only GET operations in a cross-site request.

None

Specifies that the existing behavior of sending stored cookies in requests initiated from same-site or cross-site websites continues. If this value is set to **none**, **UseSecureCookies** ACO parameter must be set to **yes** as Chrome 80 mandates secure cookies. This combination is represented as **None + Secure** in this document wherever applicable.

Use **None** to allow POST operations along with GET requests in a cross-site request. Also, set the **UseSecureCookies** ACO parameter to **yes** and ensure that the applications use secure connection.

5. Save the changes.

Configure Cookie Provider Agents to Work with SameSite Cookie Attribute

Consider the following environment:

- Agent1 is configured with cookie provider configuration in abc.com
- Agent2 acts as cookie provider in def.com

If the SiteMinder solution is not deployed on Agent2, the request from Agent1 to Agent2 fails due to the default behavior of Google Chrome 80. To resolve this issue, SiteMinder solution provides the following new ACO parameter along with the SameSite ACO parameter:

getcpcookie

If enabled, ensures that the request to cookie provider seamlessly works with additional redirects in the background. If disabled, SameSite cookie attribute settings of Chrome 80 or SameSite ACO parameter (if configured) will apply.

Default: No

To ensure that the request from Agent1 to Agent2 is successful, perform the following steps:

1. Deploy the SiteMinder solution on Agent2 that is acting as cookie provider Agent.
2. Open Administrative UI.
3. Navigate to Infrastructure, Agent, and Agent Configuration Objects.
4. Open the ACO for Agent and set the **getcpcookie** ACO parameter to **Yes**.
5. If cookie provider Agents also serve requests, set SameSite to **None** and UseSecureCookies to **Yes**.
6. Save the changes.
7. On Agent 1, set the **EnableCookieProvider** ACO parameter to **Yes** and save the change.

The following table highlights the behavior of transactions with different settings of SameSite ACO parameter on the two agents:

SameSite Value on Agent 1	SameSite Value on Agent 2	Request Type	Behavior
Strict	Strict	GET	Transaction fails
		POST	Transaction fails
Lax	Lax	GET	Transaction succeeds

SameSite Value on Agent 1	SameSite Value on Agent 2	Request Type	Behavior
		POST	Transaction fails To make this a success, configure getcpcookie on Agent 2.
None + Secure	None + Secure	GET	Transaction succeeds
		POST	Transaction succeeds
None (Default behavior of Chrome 80)	None (Default behavior of Chrome 80)	GET	Transaction succeeds
		POST	Transaction fails

Recommended Settings for Impacted Use Cases

For the list of use cases that are impacted by the default behavior of Google Chrome 80, as mentioned in the above sections, you can configure the SameSite ACO parameter to the recommended setting to make the use cases work.

The following table lists these recommended settings:

Impacted Use Case	Resolution
Non-federation	
Cookie provider flow for any POST request to an application	1. Deploy the binaries on cookie provider Agents. 2. Set **getcpcookie** to **Yes**. 3. Set **EnableCookieProvider** to **Yes**. 4. (If cookie provider Agents serve requests) Set **SameSite** to **None** and **UseSecureCookies** to **Yes**.

Impacted Use Case	Resolution
Custom cookies that are generated using responses when a POST request is initiated from a cross-site	1. Deploy the binaries on Agents. 2. Set the **SameSite** ACO parameter to **None**. 3. Set **UseSecureCookies** to **Yes**.
Auditing of Anonymous authentication scheme may result in inconsistent results	1. Deploy the binaries on Agents. 2. Set the **SameSite** ACO parameter to **None**. 3. Set **UseSecureCookies** to **Yes**.
SSO between applications when a SMSESSION cookie exists and a POST request is initiated from cross-site	1. Deploy the binaries on Agents. 2. Set the **SameSite** ACO parameter to **None**. 3. Set **UseSecureCookies** to **Yes**.
SSO between applications when a session scheme mini cookie exists and a POST request is initiated from cross-site	1. Deploy the binaries on Agents. 2. Set the **SameSite** ACO parameter to **None**. 3. Set **UseSecureCookies** to **Yes**.
Basic authentication scheme when a non-idempotent request like POST request is initiated from cross-site	No solution is currently available. Use alternative authentication schemes.
Federation	
SP-initiated federation SSO transactions with the RelayState query parameter in SAML 2.0	1. Deploy the binaries on Agents. 2. Set the **SameSite** ACO parameter to **None**. 3. Set **UseSecureCookies** to **Yes**.
SAML 2.0 SLO with HTTP-POST binding	1. Deploy the binaries on Agents. 2. Set the **SameSite** ACO parameter to **None**. 3. Set **UseSecureCookies** to **Yes**.

Impacted Use Case	Resolution
SiteMinder acts as IdP in SAML 2.0 and Authentication Mode is set to Credential Selector (CHS)	If SiteMinder is used for Delegated Authentication: 1. Deploy the binaries on Agents. 2. Set the **SameSite** ACO parameter to **None**. 3. Set **UseSecureCookies** to **Yes**. If not, the behavior depends on the third-party.
ForceAuthn in SAML 2.0 when SiteMinder acts as IdP and Authentication Requesting Binding is HTTP-POST	1. Deploy the binaries on Agents. 2. Set the **SameSite** ACO parameter to **None**. 3. Set **UseSecureCookies** to **Yes**.
Logout at IdP when SiteMinder acts as IdP and SP in a WS-FED SLO	1. Deploy the binaries on Service Provider Agents. 2. Set the **SameSite** ACO parameter to **None**. 3. Set **UseSecureCookies** to **Yes**.
AJAX requests for Single Page Applications in OpenID Connect	1. Deploy the binaries on Agents. 2. Set the **SameSite** ACO parameter to **None**. 3. Set **UseSecureCookies** to **Yes**.
Re-authentication at Authorization Endpoint (prompt=login) when SiteMinder acts as OpenID Connect Provider	1. Deploy the binaries on Agents. 2. Set the **SameSite** ACO parameter to **None**. 3. Set **UseSecureCookies** to **Yes**.

Behavior of SameSite ACO Parameter Configuration

This section describes how SameSite ACO Parameter in SiteMinder works with different settings in different scenarios.

Accessing Access Gateway Administrative UI

If **UseSecureCookies** ACO parameter is set to **Yes** and **SameSite** ACO parameter is set to **None**, ensure that SSL is enabled on Tomcat to successfully access Access Gateway Administrative UI.

SameSite and CookieDomainScope Dependencies

SameSite cookie attribute of Google Chrome 80 is not dependent on the cookie domain attribute value. SameSite ACO parameter in SiteMinder is not dependent on the value of CookieDomainScope ACO parameter as the SameSite value of a host is considered but not the value of CookieDomainScope.

SDKs

While creating a cookie using SDK, append the SameSite ACO parameter and configure it use any of the supported values.

Example: Generating a cookie using SDK

```
String ssoStringToken = ssoToken.toString();

StringBuffer cookieString = new StringBuffer();

cookieString.append("SMSESSION").append("=").append(ssoStringToken);

cookieString.append("; Domain").append("=").append("sampledomian.com");

cookieString.append("; Path").append("=").append("/");

cookieString.append("; SameSite=").append("lax");  // Use strict, lax, or none

response.addHeader("Set-Cookie",cookieString.toString());
```

Logging

To know whether the SiteMinder solution is deployed on Agents, check the Agent error log for the following line:

SameSite Fix Applied

Known Issues

The following known issues exist in this solution in cross-site requests:

- IBM WebSphere does not support SameSite cookie attribute. So, SiteMinder does not provide a solution for ASA Agent for WebSphere.

  As a workaround, consider using pre-80 Chrome or using alternative web browsers that do not mandate using SameSite cookie attribute.

- If cookie attribute contains special characters, the cookie attribute cannot be consumed by WebLogic and ServletExec application servers for Web Agent Option Pack.

- SiteMinder solution is currently unavailable for z/OS operating system.