

CyberSource Device Fingerprinting Guide

December 2019



CyberSource Contact Information

For general information about our company, products, and services, go to <http://www.cybersource.com>.

For sales questions about any CyberSource Service, email sales@cybersource.com or call 650-432-7350 or 888-330-2300 (toll free in the United States).

For support information about any CyberSource Service, visit the Support Center at <http://www.cybersource.com/support>.

Copyright

© 2019 CyberSource Corporation. All rights reserved. CyberSource Corporation ("CyberSource") furnishes this document and the software described in this document under the applicable agreement between the reader of this document ("You") and CyberSource ("Agreement"). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by CyberSource. CyberSource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of CyberSource.

Restricted Rights Legends

For Government or defense agencies. Use, duplication, or disclosure by the Government or defense agencies is subject to restrictions as set forth the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and in similar clauses in the FAR and NASA FAR Supplement.

For civilian agencies. Use, reproduction, or disclosure is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Software Restricted Rights clause at 52.227-19 and the limitations set forth in CyberSource Corporation's standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States.

Trademarks

Authorize.Net, eCheck.Net, and The Power of Payment are registered trademarks of CyberSource Corporation. CyberSource, CyberSource Payment Manager, CyberSource Risk Manager, CyberSource Decision Manager, and CyberSource Connect are trademarks and/or service marks of CyberSource Corporation.

All other brands and product names are trademarks or registered trademarks of their respective owners.

Contents

Recent Revisions to This Document 6

About This Guide 7

Audience and Purpose 7

Scope 7

Conventions 8

Note, Important, and Warning Statements 8

Text and Command Conventions 8

Related Documents 9

Customer Support 9

Chapter 1 Implementing Device Fingerprinting 10

Introduction to Device Fingerprinting 10

Elements of Device Fingerprinting 10

Device Fingerprints 10

Smart IDs 10

How Device Fingerprinting Works 11

Enhanced Profiling 11

Enhanced Profiling via Hosted SSL 12

Web Site Implementations 16

Adding the Fingerprinting Code to Your Web Site 16

Supported Tag Deployments 17

Mobile Implementations 19

Implementing the Device Fingerprinting SDK in Android Applications 19

21

Android Code Example 21

Android Return and Error Codes 22

Implementing the Device Fingerprinting SDK in iOS Applications 24

iOS Code Examples 26

iOS Return and Error Codes 26

	Specifying the Session ID in CyberSource API Requests	27
	Specifying the session_id Value	27
	Simple Order API Request Examples	28
	SCMP API Request Example	29
	Testing Your Implementation	29
Chapter 2	Configuring Custom Decision Manager Rules, Lists, and Velocity Rules	30
	Device Fingerprinting Order Elements	30
	Custom Rule Examples	31
	Screening for Suspicious Device Fingerprints	31
	Screening for Disabled Browser Attributes	32
	Screening for Device Type	33
	Screening for IP Address Characteristics	34
	Custom Fields and Lists	35
	Global Velocity	35
	Order and Product Velocity	35
Chapter 3	Reviewing Orders in Decision Manager	36
	Case Search	36
	Case Management Details	36
	Device Fingerprint Details	37
	Available Actions	39
	Similar Searches	39
	Customer Lists	39
	Information Codes	39
Appendix A	API Fields and Information Codes	40
	Simple Order API	40
	Request Fields	40
	Reply Fields	42
	Simple Order API Request and Reply Examples	50
	SCMP API	51
	Request Fields	51
	Reply Fields	52
	SCMP API Request and Reply Examples	59
	Information Codes	60
	Global Velocity	60
	Suspicious Data Information Codes	60
	Excessive Digital Identity Changes	61
	Excessive Customer Identity Changes	62

Appendix B [Code Examples for Mobile Implementation](#) 63[Android Code Example](#) 63[iOS Code Example](#) 66

Appendix C [Device Fingerprinting Cookie FAQ](#) 70

Recent Revisions to This Document

Release	Changes
December 2019	Updated to include information about Fraud Management Essentials, a simplified risk solution based on Decision Manager. Also updated procedures for the CyberSource Business Center to incorporate the new interface design.
April 2018	Updated the profiling tag examples and added the JavaScript code example for clarification for web site device fingerprinting. See "Web Site Implementations," page 16 .
March 2018	<ul style="list-style-type: none"> Added information about the CyberSource APIs to the CyberSource web site. See the CyberSource API Versions page. Added new information about Enhanced Profiling. See "Enhanced Profiling," page 11. Updated implementation steps to support new versions of device fingerprinting SDKs for iOS (v5.0-32) and Android (v5.0-96) applications. See "Web Site Implementations," page 16, and "," page 18. Updated the error codes. See "iOS Return and Error Codes," page 26. Added customer list information codes. See "Information Codes," page 60.
May 2017	<ul style="list-style-type: none"> Added true IP address state. See "Device Fingerprinting Order Elements," page 30. Added afsReply_deviceFingerprint_trueIPAddressState and score_device_fingerprint_true_ipaddress_state reply fields. See Appendix A, "API Fields and Information Codes," on page 40. Corrected Simple Order API reply field afsReply_deviceFingerprint_trueIPCity to afsReply_deviceFingerprint_trueIPAddressCity. See Appendix A, "API Fields and Information Codes," on page 40.
March 2017	<ul style="list-style-type: none"> Updated implementation steps to support new versions of device fingerprinting SDKs for iOS (v4.0-79) and Android (v4.0-90) applications. See "Web Site Implementations," page 16, and "," page 18. Added new return and error codes. See "Android Return and Error Codes," page 22, and "iOS Return and Error Codes," page 26.
October 2016	Added enhanced profiling description. See "How Device Fingerprinting Works," page 11 .

About This Guide

Audience and Purpose

This guide describes how to implement *device fingerprinting* on your web site or in your mobile applications. Device fingerprinting is a method of collecting sets of unique and non-unique identifiers that enable you to detect identity morphing, the true location of a device, and the browsing habits of individuals.

The audience for this guide includes:

- Web developers and mobile application developers who modify the check-out page of your company's web site or who develop mobile applications that your customers use to purchase merchandise from you on their phones or tablets. Mobile application developers should have either Android or iOS platform application programming skills.
- Web administrators who manage the web server.
- Software developers who add API fields to transaction requests and replies and who write the software code that integrates CyberSource services with your company's order management system.
- Decision Manager or Fraud Management Essentials (FME) administrators or case management administrators who are responsible for creating Decision Manager profiles and rules that use device fingerprints and Smart IDs to filter transactions.
- Case reviewers who use Decision Manager or FME to review orders. Reviewers can search on device fingerprints to obtain more information about customers' identities and the device that they used to place their orders.

Scope

This guide narrowly focuses on implementing and using device fingerprints and Smart IDs. For information about implementing other CyberSource services and about using Decision Manager in the Business Center, see ["Related Documents," page 9](#).

Conventions

Note, Important, and Warning Statements



Note

A *Note* contains helpful suggestions or references to material not contained in the document.



Important

An *Important* statement contains information essential to successfully completing a task or learning a concept.



Warning

A *Warning* contains information or instructions, which, if not heeded, can result in a security risk, irreversible loss of data, or significant cost in time or revenue or both.

Text and Command Conventions

Convention	Usage
bold	<ul style="list-style-type: none"> Field and service names in text. For example: Include the ics_applications field. Items that you are instructed to act upon. For example: Click Save.
<i>italic</i>	<ul style="list-style-type: none"> Filenames and pathnames. For example: Add the filter definition and mapping to your <i>web.xml</i> file. Placeholder variables for which you supply particular values.
<code>screen text</code>	<ul style="list-style-type: none"> XML elements. Code examples and samples. Text that you enter in an API environment. For example: Set the davService_run field to <code>true</code>.

Related Documents

- *Decision Manager Developer Guide Using the Simple Order API* describes how to integrate Decision Manager, a fraud detection service, with your order management system by using the Simple Order API. ([PDF](#) | [HTML](#))
- *Decision Manager Developer Guide Using the SCMP API* describes how to integrate Decision Manager, a fraud detection service, with your order management system by using the SCMP API. ([PDF](#) | [HTML](#))



The SCMP API is a legacy name-value pair API that is supported for merchants who have already implemented it. If you are new to CyberSource and want to connect to services, use the [Simple Order API](#).

- *Decision Manager User Guide* describes how to use Decision Manager in the Business Center. ([PDF](#) | [HTML](#))
- *Decision Manager Score Builder Guide* describes how to configure custom profile scores to support your business requirements. ([PDF](#) | [HTML](#))
- The [CyberSource API Versions page](#) provides information about the CyberSource API versions.

Refer to the Support Center for complete CyberSource technical documentation:

http://www.cybersource.com/support_center/support_documentation

Customer Support

For support information about any CyberSource service, visit the Support Center:

<http://www.cybersource.com/support>

Implementing Device Fingerprinting

Introduction to Device Fingerprinting

CyberSource Device Fingerprinting service gathers information about the devices that are used to place orders on your web site or about devices that use your mobile application. This information gathering process is called *device profiling*.

Elements of Device Fingerprinting

Device Fingerprints

The device fingerprint, which results from device profiling, is a unique set of identifiers derived from persistent cookies set during device profiling. This device identifier can be the single constant element that you use to detect identity morphing and the true location of a device. When identity morphing occurs, customer and transaction order data might appear to be random and derived from different customers, but the device fingerprint does not change. This fingerprint indicates that the transactions originate from a single device.

Fingerprints enable you to identify many characteristics of a device, for example:

- Connections between accounts and other customer data
- True locations of devices when they are hidden behind a proxy
- Suspicious configurations of devices, such as language settings inconsistent with the country

Smart IDs

Unlike device fingerprints, Smart IDs are not based on cookies. You can use them to detect the browsing patterns of customers who delete cookies, use private browsing mode, or steal cookies from other users. In rare situations it is possible for two devices, especially mobile devices, to have the same Smart ID.

Smart IDs have a lifetime of approximately two weeks starting from the first time that a device visits a tagged web page or mobile application. For example, if a device with a Smart ID visits a tagged web site once, but does not visit a tagged web site again for 3 weeks, the device receives a new Smart ID the next time it visits a tagged web site.

However, if that device visits a tagged web site one day, and then visits another tagged web site or the same tagged web site within approximately 14 days, that Smart ID might persist. As long as the device user remains active on tagged pages without lapses that exceed the Smart ID's lifetime, the Smart ID persists and its lifetime is extended each time the customer visits a tagged page.

Smart IDs are based on device attributes. As a result, if critical elements of the device change, the Smart ID may also change. Examples of critical elements include enabled browser elements (JavaScript, flash, cookies, or images), operating system, and browser plug-ins.

How Device Fingerprinting Works

- 1 You add the device fingerprinting code to your web site, or you add the device fingerprinting code and libraries to your mobile application.
- 2 A customer opens a page on your web site with their browser or launches your mobile application, and the code you inserted in Step 1 sends information about their device to the device fingerprinting server along with a unique session ID that identifies the session.
- 3 The device fingerprinting server profiles the device. This profiling process collects device identification information.
- 4 You send an API request to the CyberSource server that contains the same session ID that is sent to the fingerprinting server in Step 2.
- 5 The CyberSource server returns information from the device profiling performed in Step 3 to Decision Manager or Fraud Management Essentials, which you can use to determine whether the transaction is legitimate or fraudulent.

Enhanced Profiling

Device fingerprint collection technology uses the connection between the customer's device and the merchant's shopping cart. Because dozens of browsers are supported on hundreds of device platforms, which often change regularly, there may be instances in which aspects of the device fingerprint are not reliable or present during a transaction. This could be due to a customer explicitly blocking device collection methods or certain default browser behaviors—all which are out of a merchant's control.

Certain network-based enhancements can improve aspects of device fingerprint collection. One such enhancement is Enhanced Profiling, which implements SSL certificates to make all browser communication and web objects appear to be from the merchant's domain. Enhanced Profiling increases the accuracy of device profiling, mitigates third party cookie blocking, substantially reduces end-user's ability to filter out profiling code, and increases your confidence in blocking transactions from unprofiled devices.

CyberSource can manage the SSL process, which has an associated cost. For more information, contact your CyberSource account manager.



Enhanced profiling will require custom integration and additional billing for Fraud Management Essentials.

Enhanced Profiling via Hosted SSL

CyberSource recommends that all customers using Device Fingerprint use the Enhanced Profiling feature. It increases the accuracy of device profiling and reduces the visitor's potential concern about third-party content on your website. With the enhanced profiling service, all profiling requests from the visitor's browser or native mobile application are made to a domain that is secured by the customer's SSL digital certificates.

In order to profile a device, profiling tags are placed on your website, or the Device Fingerprint SDK is embedded in your mobile application. The tags or SDK request objects from and transmit data to the device fingerprinting profiling server. Because the CyberSource-based profiling server is a different domain than the original domain from which the initial page or mobile application typically requests resources, visitors who are monitoring the network requests can see that the requests are being made to a third-party server.

Enhanced Profiling prevents the blocking of third-party requests, either natively as a feature of the browser or through third-party browser privacy tools.



Enhanced Profiling is required for device fingerprinting mobile implementations and highly recommended for web site implementations.

Steps for Setting Up Enhanced Profiling

Setting up Enhanced Profiling involves these steps:

- 1 Choose a host name.
- 2 Provide [required and/or optional fields](#) to your CyberSource services representative.
- 3 CyberSource generates a certificate signing request (CSR) and provides it to you.
- 4 Have the CSR signed by your preferred Certificate Authority (CA).
- 5 Return the signed certificate, root certificate for your CA, and the chain (or bundle) file to CyberSource.
- 6 CyberSource supplies the dedicated host name that is assigned for your profiling server.

- 7 Deploy the SSL certificate to the production environment.
- 8 Configure your DNS.
- 9 Update your profiling tags to use the new host name.

The deployment timeframe for SSL certificates is 5 to 7 days upon receipt of the SSL certificate, bundle certificates, and root certificate authority (CA). All of these file are required to minimize the deployment time of the SSL certificates.

See the following sections ["Selecting a Sub-Domain Name for Your Profiling Server," page 13](#), and ["Obtaining the CSR for Your Profiling Server Host Name and a Signed Certificate File," page 14](#), for further details on submitting SSL requests. Contact CyberSource Customer Support if you have questions.

Selecting a Sub-Domain Name for Your Profiling Server

Choose a host name that does not indicate that the server is used as a security measure. For example:

- content.yourdomain.com
- img2.yourdomain.com
- cdnA.yourdomain.com

When you have chosen a host name to use for profiling, provide it to your CyberSource services representative along with the fields described below.



Note

These fields are the minimum information needed to create a certificate signing request (CSR). Your Certificate Authority (CA) may require additional information. Check with them prior to providing this information to CyberSource to ensure that the CSR is created according to the CA requirements.

The following fields are required:

- Country name (two-letter code): US
- Organization name (for example, company) [Internet Widgits Pty Ltd.]: Lemon Bank, Inc.
- Common name or fully qualified domain name (for example, sub-domain that has been chosen): content.lemonbank.com

The following fields are optional:

- State or province name (full name) [some-state]: California
- Locality name (for example, city): San Jose
- Organizational unit name (for example, section): Services

Obtaining the CSR for Your Profiling Server Host Name and a Signed Certificate File

The following examples use *content.yourdomain.com*. If you decide on a different name, simply substitute it in the instructions outlined.

CyberSource generates a certificate signing request (CSR), which is delivered to you to be signed by your preferred Certificate Authority (CA). You return the signed certificate, root certificate for your CA, and the chain (or bundle) file to CyberSource. The certificate should be in PEM format and named in this format: *content.yourdomain.com.crt*. If the certificate requires a chain file (Intermediate CA bundle file), provide that file using the name *content.yourdomain.com_bundle.crt*. In some cases the CA may require more than one chain; verify that with the CA you chose.

In addition to supplying the CSR, CyberSource supplies the dedicated host name that is assigned for your profiling server. This unique host name is required in order for you to configure your DNS. Please note that the DNS redirection is not available until the SSL certificate is deployed into the production environment. You can direct all traffic to *content.yourdomain.com* by setting up a CNAME record in your DNS.

The entry should resemble the following example:

```
content.yourdomain.com. CNAME h-yourdomain.online-metrix.net.
```

The certificate is then installed on the Profiling Servers.



Note

If your corporate security policy does not allow generation of the CSR, contact CyberSource Customer Support to discuss alternative methods.

In the event that a wildcard certificate is deployed, a Fully Qualified Domain Name (FQDN) is required for monitoring proposes. This requires you to create an entry as outlined in the example above and provide it to CyberSource.

Updating Profiling Code to Use the New Host Name

After you receive confirmation from CyberSource that your SSL certificate is deployed to the profiling servers, you should change the host name used in the profiling tags and/or SDK to content.yourdomain.com.



You should make this change only after you verify that the certificate is deployed. Making this change before the certificate is deployed (even if the DNS changes are complete) might generate security warnings in your end-users' web browsers.

Notice to European Union Merchants

The European Union's Privacy and Electronic Communications Directive (the "Directive") restricts the deposit and storage of cookies on the devices of customers of online merchants operating in the European Union.

The device fingerprint feature of CyberSource Decision Manager and CyberSource Decision Manager Account Takeover Protection Service is one of more than two hundred global fraud detectors and tests. This feature enables the deposit and storage on the customer's computer of a cookie that profiles the specific attributes of the computer used in transactions. This cookie is used to mitigate fraud.

While we cannot provide legal advice to our merchants, we can provide the following information. The restrictions under the Directive require, among other things, that you

- Provide "clear and comprehensive information" to visitors of your web site about the storage of cookies on their computer.
- Obtain the consent of visitors before depositing and storing cookies on their computer unless certain exceptions apply.

Your compliance with applicable privacy laws depends on how you use the cookies, on what information you disclose to customers, and on what consent you obtain from customers. Because CyberSource has no direct connection to your customers, you are responsible for ensuring that cookies are used properly to perform the requested CyberSource services. CyberSource believes that the safest course of action is for you to clearly and conspicuously disclose the use of cookies to your customers and to obtain their consent before placing cookies on their devices. If you operate in Europe and use the device fingerprint, you should consult your legal counsel and other advisors to find out how to comply with the requirements of the Directive and whether an exception might be available for you. CyberSource cannot take any position on the storage of cookies on the devices of customers for purposes other than to provide CyberSource services. When used without the device fingerprint, Decision Manager does not store cookies. Decision Manager Account Takeover Protection Service requires the device fingerprint and must store cookies to operate.

Web Site Implementations

You can deploy device fingerprinting by configuring your web site as described in the following section.



Important

To ensure your customers' privacy, CyberSource encodes fingerprints as soon as they are received. Fingerprints persist for approximately 24 hours. This interval begins when the customer opens the HTML page with the tags, and it ends when the transaction request is sent to CyberSource. Add the fingerprint to your request as soon as possible.



Warning

CyberSource recommends that you use domain names instead of using IP addresses and relying on domain name resolution. Device fingerprinting stops working when the IP address of the domain name changes.

Adding the Fingerprinting Code to Your Web Site

Various parameters are specified in the profiling tag:

- **<org ID>** (mandatory): to obtain this value, contact your CyberSource representative and specify whether it is for testing or production.
- **<merchant ID>** (mandatory): your unique CyberSource merchant ID.
- **<session ID>** (mandatory): a session ID must be a unique identifier for the transaction, such as an order number. It can contain lowercase and uppercase English letters, digits, hyphens (-), and underscores (_). The maximum length is 88 characters. The session ID must be unique for each transaction and for each merchant ID. You can use any string that you are already generating, such as an order number or web session ID. Do not use the same uppercase and lowercase letters to indicate different session IDs.

The session ID must be unique for each page load regardless of an individual's web session ID. If a user navigates to a profiled page and is assigned a web session, navigates away from the profiled page, then navigates back to the profiled page, the generated session ID should be different and unique. You may use a web session ID, but it is preferable to use an application GUID (Globally Unique Identifier). This measure ensures that a unique ID is generated every time the page is loaded, even if it is the same user reloading the page.

- **Custom Profiling Domain** (optional): domain that serves the JavaScript tag. The default is `h.online-metrix.net`. As an alternative, you can implement Enhanced Profiling so that all profiling requests are made to a domain that is secured by your SSL digital certificates. See ["Enhanced Profiling," page 11](#), for more information.

Be sure to copy all characters correctly and to omit the angle brackets (< >) when substituting your values for the variables.

Supported Tag Deployments

Two methods are available to deploy the profiling tag as described in the ["Tag Placement"](#) section.

To allow device profiling time to complete, ensure that 3 to 5 seconds elapse between the execution of the profiling code and when your customers submit their orders.

Tag Placement

Place the basic `<script>` tag in the `<head>` tag for optimal performance for either method.

For the basic `<script>` tag with `<noscript>` tag method, place the `<noscript>` tag in the `<body>` tag, as in [Example 1](#). Do not place the `<noscript>` tag in the `<head>` tag because it contains an `<iframe>` tag. Placing iframes in the head tag violates W3C validation and might cause problems.

JavaScript Code

```
<head>

    <script type="text/javascript" src="https://h.online-metrix.net/fp/
tags.js?org_id=<org ID>&session_id=<merchant ID><session ID>"></
script>

</head>

<body>

    <noscript>

        <iframe style="width: 100px; height: 100px; border: 0; position:
absolute; top: -5000px;" src="https://h.online-metrix.net/fp/tags?org_
id=<org ID>&session_id=<merchant ID><session ID>"></iframe>

    </noscript>

</body>
```

Basic <script> Tag with <noscript> Tag (Recommended)

The recommended deployment includes a basic `<script>` tag, which loads a Javascript resource: `tags.js`, as well as an additional `<noscript>` tag. The purpose of the `<noscript>` tag is to ensure that profiling occurs, even if JavaScript is disabled.

Example 1 Basic <script> Tag with <noscript> Tag

```
<head>

    <script type="text/javascript" src="https://h.online-metrix.net/fp/
tags.js?org_id=sample_orgID&session_id=sample_merchantIDsample_
sessionID"></script>

</head>

<body>

    <noscript>

        <iframe style="width: 100px; height: 100px; border: 0; position:
absolute; top: -5000px;" src="https://h.online-metrix.net/fp/tags?org_
id=sample_orgID&session_id=sample_merchantIDsample_sessionID"></iframe>

    </noscript>

</body>
```

Basic <script> Tag (without <noscript> Tag)

This deployment includes a single `<script>` tag that loads a JavaScript resource: `tags.js`. If JavaScript is disabled, this tag does not load, and profiling does not occur.

Example 2 Basic <script> Tag (without <noscript> Tag)

```
<head>

    <script type="text/javascript" src="https://h.online-metrix.net/fp/
tags.js?org_id=sample_orgID&session_id=sample_merchantIDsample_
sessionID"></script>

</head>
```

Mobile Implementations

You can deploy device fingerprinting in Android and iOS applications.



Implementing device fingerprinting in mobile applications requires either Android or iOS platform application programming skills.

Implementing the Device Fingerprinting SDK in Android Applications

To implement the device fingerprinting mobile SDK for Android, you must use Android 4.0 or later.

Devices using MIPS application processors are not supported.

The SDK uses [Android's native API](#) as the default API for data transmission. You can use the OkHTTP library (3+) by passing true to the `setUseOkHTTP()` method.

You must explicitly set the time unit for all timeout settings within the SDK.



If your application supports Android API 15 or earlier, you must do one of the following:

- Add this code to your application before making any calls to the SDK:

```
System.setProperty("java.io.tmpdir",  
    getApplicationContext().getDir("files", Context.MODE_  
PRIVATE).getPath())
```

OR

- Add this permission to the manifest file of your application:

```
<uses-permission  
    android:name="android.permission.WRITE_EXTERNAL_  
STORAGE" />
```

To implement device fingerprinting in Android applications:

Step 1 Download the *CyberSourceTMDeviceFingerprintingMobileSDK_for_Android.zip* file from the Business Center Documentation page, and add it to your project.

Step 2 The *zip* file contains two different formats. Include only one of these formats in your project. The selection of files includes:

- a** *TrustDefender-<version>.aar* contains both the Java files and the Android native shared library.

The standard way of including an AAR dependency to Android is described at: <https://developer.android.com/studio/projects/android-library.html>. Refer to the "Add your library as a dependency" section.

When updating with the new AAR file, ensure that you "Clean Project" to delete the cached version of the library.

- b** *TrustDefender-<version>.zip* contains the Java files as a Jar file (*TrustDefender-<version>.jar*) and the native files as Shared Object Files (.so). This is intended for integrating with Eclipse because Eclipse doesn't recognize AAR files.

To use the zip file, simply unzip it and put the content in the *lib* directory of your project. Assuming the current path is the root of project, use the following command to unzip the ThreatMetrix SDK to the libs directory of your project:

```
unzip TrustDefender-<VERSION>.zip -d libs
```

The directory structure should appear as follows:

```
libs/armeabi/libtrustdefender-jni.so
libs/armeabi-v7a/libtrustdefender-jni.so
libs/armeabi-v8a/libtrustdefender-jni.so
libs/TrustDefender-<version>.jar
libs/x86/libtrustdefender-jni.so
libs/x86_64/libtrustdefender-jni.so
```

- c** *TrustDefender-<version>-javadoc.jar* contains the Javadoc style documentation, which may be added to the project to provide documentation within the Integrated Development Environment (provided the IDE supports it). It is not required, however, and is included only as a programming aid.

Step 3 Include the following permission in the mobile application manifest file:

```
<uses-permission android:name="android.permission.INTERNET">
</uses-permission>
```

- Step 4** Specify your merchant ID and the session ID as a concatenated value for a variable that is passed to the `TrustDefenderMobile` class in your Android application. In the following example, `my_variable`=your merchant ID + the session ID as a concatenated value:

```
profile.setSessionID ("my_variable");
```

The `TrustDefenderMobile` class is contained in the *CyberSourceTMDeviceFingerprintingMobileSDK_for_Android.zip* file. A session ID must be a unique identifier for the transaction, such as an order number. It can contain lowercase and uppercase English letters, digits, hyphens (-), and underscores (_). The maximum length is 88 characters. The session ID must be unique for each transaction and for each merchant ID. You can use any string that you are already generating, such as an order number or web session ID. Do not use the same uppercase and lowercase letters to indicate different session IDs.

The session ID must be unique for each page load, regardless of an individual's web session ID. If a user navigates to a profiled page and is assigned a web session, navigates away from the profiled page, then navigates back to the profiled page, the generated session ID should be different and unique. You may use a web session ID, but it is preferable to use an application GUID (Globally Unique Identifier). This measure ensures that a unique ID is generated every time the page is loaded, even if it is the same user reloading the page.

- Step 5** Add the `doProfileRequest()` function to your application, and specify the following required calling options:

Option	Description
Org ID	Contact CyberSource Customer Support for this value and specify whether it is for testing or production.
Fingerprint server URL	Fully qualified domain name (FQDN) of the server. It must be specified in an FQDN format, for example, <i>host.domain.com</i> , NOT in URL format.

Android Code Example

See [Appendix B, "Code Examples for Mobile Implementation," on page 63](#) for the Android code example.

After you add the device fingerprinting mobile SDK to your application, you must specify the session ID in the API request that you send to CyberSource by using the `deviceFingerprintID` Simple Order API request field or the `device_fingerprint_id` SCMP API request field.

Android Return and Error Codes

The following table lists the codes you may encounter when implementing the Device Fingerprinting SDK in an Android application.



The return profiling code **THM_OK** must be present before you send the API request. This code ensures the presence of a complete profile.

Table 1 Android Return and Error Codes

Value	Description
THM_NotYet	The profiling request is not yet complete.
THM_OK	Device profiling completed with no errors.
THM_Connection_Error	A connection issue was encountered between the device and the fingerprint online server. Ensure that you are referencing the server correctly and that you can access it from the device.
THM_HostNotFound_Error	The host name of the fingerprint server could not be resolved. Ensure that you are referencing the server correctly and that you can access it from the device.
THM_NetworkTimeout_Error	A timeout occurred while the device was communicating with the fingerprint server. A timeout can occur if the device's internet connection is disabled while it is communicating with the server.
THM_HostVerification_Error	The fingerprint server host name in use does not match the host name in the certificate, which may be evident when you use the Advanced Profiling feature, or implement in a proxied scenario with the custom URL option. Ensure that a valid certificate is in use on the target host name specified in the custom URL option.
THM_Internal_Error	A miscellaneous error was detected. Check the input/options used when calling the library.
THM_Interrupted_Error	The profiling request was interrupted or canceled mid-flight.
THM_InvalidOrgID	This code is returned if an invalid or NULL value is present in the org_id calling option.
THM_PartialProfile	A connection error resulted in partial profiling.

Table 1 Android Return and Error Codes (Continued)

Value	Description
THM_Blocked	<p>The profiling request cannot be processed because profiling is blocked due to some conditions. The most common scenario is that the phone screen is off longer than the amount of time specified by the screenOffTimeout value. You can customize this value by calling the Config.setScreenOffTimeout() method during init().</p> <pre>Config config = new Config() .setContext(getApplicationContext()) .setScreenOffTimeout(180); profile.init(config);</pre>
THM_ConfigurationError	Mobile SDK is not activated for the customer.
THM_Already_Initialised	SDK is already initialized; use the current instance.
THM_EndNotifier_Not_Found	EndNotifier is not passed to doProfileRequest , and it is mandatory in the profile request.
THM_ThirdPartyLibrary_Not_Found	<p>OkHttp library is not available; include the library. For information about how to include the library see http://square.github.io/okhttp/</p> <p>Note Both OkHttp 2 and OkHttp 3 are supported.</p>

Implementing the Device Fingerprinting SDK in iOS Applications

To develop iOS applications, you must be enrolled in the iOS Developer Program, which enables you to upload your applications to the Apple App Store. To link to the CyberSource device fingerprinting mobile SDK, you must use the iOS 8 or later and the Apple Xcode 6.0 IDE.

To implement device fingerprinting in iOS applications:

Step 1 Download the *CyberSourceTMDeviceFingerprintingMobileSDK_for_iOS.zip* file from the Business Center Documentation page, and add it to your project.

Step 2 Import the device fingerprinting SDK libraries and frameworks into your iOS application:

```
#import <TrustDefender/TrustDefender.h>
```

- If using Modular Import, add the following code as needed to your project:

```
@import TrustDefender; //Objective-C
```

Or

```
import TrustDefender //Swift
```

Step 3 Link your framework(s):

- If using Modular Import, link the following framework:
 - zlib (libz.dylib)
- If not using Modular Import, link the following frameworks:
 - Security
 - UIKit
 - Foundation
 - CoreTelephony
 - CoreLocation
 - zlib (libz.dylib)



Note

For Modular Import:

- Autolinking does not work because the SDK is a static framework. Therefore, you must add the framework to your project manually.
 - Ensure that the "Enable Modules (C and Objective-C)" option in the project setting is set to "YES."
 - Remember to add the -ObjC linker flag in Xcode; otherwise, SystemConfiguration and CoreTelephony are not linked automatically.
-

For information about linking to libraries and frameworks in iOS applications, see:

https://developer.apple.com/library/ios/recipes/xcode_help-project_editor/Articles/AddingALibrarytoATarget.html

- Step 4** Specify your merchant ID and the session ID as a concatenated value for a variable that is passed to the `TrustDefenderMobile` class in your iOS application. In the following example, **`my_variable`** = your merchant ID + the session ID as a concatenated value:

```
self.profile.sessionID = @"my_variable" ;
```

The `TrustDefenderMobile` class is contained in the *CyberSourceTMDeviceFingerprintingMobileSDK_for_iOS.zip* file. A session ID must be a unique identifier for the transaction, such as an order number. It can contain lowercase and uppercase English letters, digits, hyphens (-), and underscores (_). The maximum length is 88 characters. The session ID must be unique for each transaction and for each merchant ID. You can use any string that you are already generating, such as an order number or web session ID. Do not use the same uppercase and lowercase letters to indicate different session IDs.

The session ID must be unique for each page load, regardless of an individual's web session ID. If the same user navigates to a profiled page and is assigned a web session, navigates away from the profiled page, then navigates back to the profiled page, the generated session ID should be different and unique. You may use a web session ID, but it is preferable to use an application GUID (Globally Unique Identifier). This measure ensures that a unique ID is generated every time the page is loaded, even if it is the same user reloading the page.

- Step 5** Add the `doProfileRequest()` function to your application, and specify the following calling options:

Option	Description
Org ID	Contact CyberSource Customer Support for this value and specify whether it is for testing or production.
Fingerprint server URL	Fully qualified domain name (FQDN) of the server. It must be specified in an FQDN format, for example, <i>host.domain.com</i> , NOT in URL format.

To fix "selector not recognized" runtime exceptions when trying to use category methods from a static library:

- Step 1** See the following article for more information:

https://developer.apple.com/library/mac/qa/qa1490/_index.html

iOS Code Examples

See [Appendix B, "Code Examples for Mobile Implementation,"](#) on page 63 for the iOS code examples.

After you add the device fingerprinting mobile SDK to your application, you must specify the session ID in the API request that you send to CyberSource by using the [deviceFingerprintID](#) Simple Order API request field or the [device_fingerprint_id](#) SCMP API request field.

iOS Return and Error Codes

The following table lists the codes you may encounter when implementing the Device Fingerprinting SDK in an iOS application.



Important

The return profiling code **THMStatusCodeOk** must be present before you send the API request. This code ensures the presence of a complete profile.

Table 2 iOS Return and Error Codes

Value	Description
THMStatusCodeNotYet	The profiling request is not yet complete.
THMStatusCodeOk	Device profiling completed with no errors.
THMStatusCodeConnectionError	A connection issue was encountered between the device and the fingerprint online server. Ensure that you are referencing the server correctly and that you can access it from the device.
THMStatusCodeHostNotFoundError	The host name of the fingerprint server could not be resolved. Ensure that you are referencing the server correctly and that you can access it from the device.
THMStatusCodeNetworkTimeoutError	A timeout occurred while the device was communicating with the fingerprint server. A timeout can occur if the device's internet connection is disabled while it is communicating with the server.
THMStatusCodeHostVerificationError	The fingerprint server host name in use does not match the host name in the certificate, which may be evident when you use the Advanced Profiling feature or implement in a proxied scenario with the custom URL option. Ensure that a valid certificate is in use on the target host name specified in the custom URL option.
THMStatusCodeInternalError	A miscellaneous error was detected. Check the input/options used when calling the library.
THMStatusCodeInterruptedError	The profiling request was interrupted or canceled mid-flight.

Table 2 iOS Return and Error Codes (Continued)

Value	Description
THMStatusCodePartialProfile	A connection error resulted in partial profiling.
THMStatusCodeInvalidOrgID	This code is returned if an invalid or NULL value is present in the org_id calling option.
THMStatusCodeNotConfigured	This code is returned if the object is not configured properly. You may receive this code if you pass the wrong options to the configure method.
THMStatusCodeCertificateMismatch	This code is returned when there is a mismatch between the certificate pinned and the certificate used in the host.

Specifying the Session ID in CyberSource API Requests

After you add the device fingerprinting code to your web site or mobile application, you must specify the session ID in Decision Manager or Fraud Management Essentials transactions by using the [deviceFingerprintID](#) Simple Order API request field or the [device_fingerprint_id](#) SCMP API request field. If you do not include this API request field along with the other API request fields in the transaction request, no device fingerprinting information is returned in the reply.

After you specify the session ID in your API request, you can test your implementation. See ["Testing Your Implementation," page 29](#).

Specifying the session_id Value

The syntax used to specify the `session_id` value for web pages and mobile applications differs from that used with the API field:

- In web pages and mobile applications, use `session_id=<merchant id><session ID>` where your merchant ID is concatenated with the session ID.
- In API requests, use the [deviceFingerprintID](#) (Simple Order API) or [device_fingerprint_id](#) (SCMP API) field to specify the `<session ID>`.

Simple Order API Request Examples

For more examples, see ["Simple Order API Request and Reply Examples,"](#) page 50.

Example 3 Simple Order API Name-Value Pair

```

afsService_run=true
<customer's name and billing address fields>
card_accountNumber=4111xxxxxxxx1111
card_cardType=001
card_expirationMonth=12
card_expirationYear=2018
cc_AuthService_run=true
deviceFingerprintID=5834125431628311477
merchantDefinedData_mddField32=126
merchantID=example
merchantReferenceCode=833617922960995060
purchaseTotals_currency=USD
purchaseTotals_grandTotalAmount=30.00

```

Example 4 Simple Order API XML

```

<requestMessage xmlns="urn:schemas-cybersource-com:transaction-
dataschema_version_number">
  <merchantID>example</merchantID>
  <merchantReferenceCode>833617922960995060</merchantReferenceCode>
  <billTo>
    <customer's name and billing address fields>
  </billTo>
  <purchaseTotals>
    <currency>USD</currency>
    <grandTotalAmount>30.00</grandTotalAmount>
  </purchaseTotals>
  <card>
    <accountNumber>4111xxxxxxxx1111</accountNumber>
    <cardType>001</cardType>
    <expirationMonth>12</expirationMonth>
    <expirationYear>2018</expirationYear>
  </card>
  <merchantDefinedData>
    <mddField id="32">126</mddField>
  </merchantDefinedData>
  <afsService run="true">
  <ccAuthService run="true">
  <deviceFingerprintID>5834125431628311477</deviceFingerprintID>
</requestMessage>

```

SCMP API Request Example

Only name-value pairs are supported in the SCMP API. For more examples, see "[SCMP API Request and Reply Examples](#)," page 59.

Example 5 SCMP API Name-Value Pair

```
ics_applications=ics_score
<customer's name and billing address fields>
customer_cc_number=4111xxxxxxxx1111
card_type=001
customer_cc_expmo=12
customer_cc_expyr=2018
ics_applications=ics_auth
device_fingerprint_id=5834125431628311477
merchant_defined_data32=126
merchant_id=example
merchant_ref_number=833617922960995060
currency=USD
grand_total_amount=30.00
```

Testing Your Implementation

To test your implementation:

- Step 1** Create a custom rule to screen orders for the presence of a fingerprint.
 - Step 2** Send a test API request. Your test reply contains a fingerprint if your implementation is correct.
-

Configuring Custom Decision Manager Rules, Lists, and Velocity Rules

You can use the device fingerprinting attributes that are returned in the API reply to configure rules for order profiles.



Important

With the exception of device velocity and certain standard rules that make use of device details like IP Address, Fraud Management Essentials does not permit custom rules, lists or velocity rules. This chapter is not relevant to Fraud Management Essentials.

Device Fingerprinting Order Elements

This table lists device fingerprinting order elements that are available in the Rule Editor:

Table 3 Available Device Fingerprinting Order Elements

■ Application type	■ Profiled URL
■ Browser language	■ Proxy IP address
■ Cookies enabled	■ Proxy IP address activities
■ Device fingerprint	■ Proxy IP address attributes
■ Device latitude	■ Proxy server type
■ Device longitude	■ Screen resolution
■ Device matched	■ Smart ID
■ Flash enabled	■ Smart ID confidence level
■ Flash operating system	■ Time on page
■ Flash version	■ True IP address
■ GPS accuracy	■ True IP address activities
■ Images enabled	■ True IP address attributes
■ Jailbreak/root privileges	■ True IP address city
■ Jailbreak/root reason	■ True IP address state
■ JavaScript enabled	■ True IP address country
■ Profiling duration	

For other order elements that are available in the Rule Editor, see Appendix A, “Custom Rules Elements and Examples,” in the *Decision Manager User Guide* ([PDF](#) | [HTML](#)).

Custom Rule Examples

Screening for Suspicious Device Fingerprints

You can create custom rules that specify identity, suspicious, and velocity information codes that can be returned in replies. This example shows a rule that screens orders for a fingerprint that was already deemed suspicious. If the rule is triggered, the third condition increases the probability that the order is fraudulent. For a complete list of Fraud Score order elements, see Appendix A in the *Decision Manager User Guide* ([PDF](#) | [HTML](#)).

Example 6 Rule That Screens for Suspicious Device Fingerprint

First condition	
Order element	Fraud score suspicious information
Comparison operator	contains
Comparison values	Device confirmed risky
Second condition	
Order element	Fraud score customer list information
Comparison operator	contains
Comparison value	Device fingerprint on negative list
Third condition	
Order element	Fraud score suspicious information
Comparison operator	contains
Comparison values	Masked device history
Condition relationship	At least one condition is true.
Profile Setting	Reject orders that contain a true condition.

Screening for Disabled Browser Attributes

This example shows a rule that triggers a review of orders when a customer disables browser attributes, which might indicate suspicious activity. This example contains all possible elements that can be detected as disabled in customers' browsers. However, your rule might contain only those that you consider most likely to reveal suspicious activity for your business.

Example 7 Rule That Screens for Disabled Browser Attributes

First condition	
Order element	Cookies enabled
Comparison operator	is equal to
Comparison values	false
Second condition	
Order element	Flash enabled
Comparison operator	is equal to
Comparison value	false
Third condition	
Order element	Images enabled
Comparison operator	is equal to
Comparison values	false
Fourth condition	
Order element	JavaScript enabled
Comparison operator	is equal to
Comparison values	false
Condition relationship	All conditions are true.
Profile setting	Review or reject orders that contain all true conditions.

Screening for Device Type

This example shows a rule that helps you to discover the type of device used to place the order, such as a mobile phone.

Example 8 Rule That Screens for Device Type

First condition

Order element	Application type
Comparison operator	is equal to
Comparison values	Custom value (for example: <code>browser_mobile</code>)

Second condition

Order element	Device latitude
Comparison operator	is present

Third condition

Order element	GPS accuracy
Comparison operator	is present

Condition relationship At least one condition is true.

Profile setting Review orders that trigger the rule.

Screening for IP Address Characteristics

This example shows a rule for screening orders for the suspicious attributes and activities of a proxy IP address. You must create one condition for each comparison value that you choose.

Example 9 Rule That Screens for IP Address Characteristics

First set of conditions

Order element	Proxy IP address activities
Comparison operator	contains
Comparison values	Phishing
	Nigerian email or spam
	UDP port scan
	TCP port scan
	Connecting to botnet
	Connecting to malware site
	Connecting to suspicious IRC server
	Click fraud
	Malware
	Spam

Second set of conditions

Order element	Proxy IP address attributes
Comparison operator	contains
Comparison values	Bogon
	Hijacked
	Open relay
	Zombie or botnet

Condition relationship	At least one condition is true.
-------------------------------	---------------------------------

Profile setting	Review or reject orders that contain a true condition.
------------------------	--

Custom Fields and Lists

You can customize rules with any operator in the condition editor. For example, you can create a list of IP addresses. You can modify the items in the list as often as necessary. After you add the list to a custom rule, you can set the profile that contains the rule to review or reject orders depending on the IP addresses found in the orders.

Global Velocity

You can track device fingerprints at specific intervals in the Business Center. An information code is returned for each test that is triggered. By default, all time intervals are checked. False-positive results might occur during high-volume shopping periods. For example, during end-of-year holidays customers might make frequent purchases within a short period of time. During this time they might ship their gift purchases to different addresses, which might trigger other rules and also produce false-positive results. For more information, see the *Decision Manager Developer Guide Using the Simple Order API* ([PDF](#) | [HTML](#)), the *Decision Manager Developer Guide Using the SCMP API* ([PDF](#) | [HTML](#)), and the *Decision Manager User Guide* ([PDF](#) | [HTML](#)).

Order and Product Velocity

To evaluate a fingerprint in relation to product, time, number, or value of orders, you can create order and product velocity rules specific to your business needs.

Reviewing Orders in Decision Manager

You can view an encoded fingerprint in the Case Management and the Transaction Search nodes of the Business Center and use it to review orders. The encoded fingerprint appears as a string ending with an equal sign (=). For example:
77a8cbfbf3d7480e8aea4869eb1ca0c0=. The fingerprint is stored in the fraud database with the rest of the transaction data for the same length of time (180 days).



Important

This chapter only applies to Decision Manager and not Fraud Management Essentials.

Case Search

To search for device fingerprints, go to the Field and value tab in the Case Search window, and choose **Device Fingerprint** from the Field list. When searching for a device fingerprint, you can specify any date range, but you cannot export the search results.

Case Management Details

If the fingerprint is available, more information might be available about the customer's identity and the device used to place the order. There are three areas of the Case Management Details window that you can use in your review process:

- Device Fingerprint link, which launches a dialog box with details about the device
- Available Actions menu, which you can use to mark the transaction
- Similar Searches menu, which you can use to search on the device fingerprint

Device Fingerprint Details

When you click the Device Fingerprint link in the Case Management Details window, the Device Fingerprint dialog box appears, which contains information about the device. There is a Smart ID link if the Smart ID is available instead of the device fingerprint. Any of these fields in the dialog box can contain information:

Table 4 Device Fingerprint Dialog Box Descriptions

Field	Description
Device Fingerprint	Unique ID of a computer or other device.
Smart ID	Device identifier generated from attributes collected during profiling. The confidence level follows the smart ID. Its value ranges from 0 to 100 and indicates the probability that the Smart ID is correctly identifying a returning device. A high percentage is more likely to represent a returning device than a new device that is similar to a previously identified device. As the confidence level decreases, the likelihood of a false positive increases.
Profiling Date/Time	Time of device profiling.
Profiling Duration	Total time in milliseconds to process the profiling request.
Profiled URL	URL of the profiled page.
Date Device First Seen	Date, in UTC, on which the device was first encountered.
Device Matched	Indicates whether the device was previously encountered and whether enough attributes were gathered to identify the device: <ul style="list-style-type: none"> ■ Success: Device fingerprint was previously encountered. ■ New_Device: Device was not previously encountered. ■ Not_Enough_Attrrib: Not enough attributes were gathered to indicate whether the device was previously encountered.
Application Type	Indicates whether the session was initiated from a mobile device or a computer. If the session is initiated from a mobile device, this field indicates whether the mobile browser or mobile application is being used: <ul style="list-style-type: none"> ■ browser_computer: Device is using a standard browser, which contains the fingerprinting tags. ■ browser_mobile: Device is using a mobile browser, which contains the fingerprinting tags. ■ agent_mobile: Device is using a mobile application, and fingerprinting mobile SDK tags are present in that mobile application.
Time on Page	Time period in milliseconds that the device profiling page appears in the browser before it closes or the user navigates away from the page.
Enabled in Browser	Indicates whether Flash, images, JavaScript, or cookies are enabled in the device.

Table 4 Device Fingerprint Dialog Box Descriptions (Continued)

Field	Description
Disabled in Browser	Indicates whether Flash, images, JavaScript, or cookies are disabled in the device.
Screen Resolution	Screen resolution of the device, which can distinguish a computer from a mobile device.
Browser Language	Language detected in the browser, such as English or Japanese.
Browser Session ID	The concatenated merchant ID and session ID value that is sent in with the request. See deviceFingerprintID , page 41, if you are using the Simple Order API, or device_fingerprint_id , page 51, if you are using the SCMP API.
Flash Operating System	Device operating system as reported by Flash.
Flash Version	The version of Flash installed on the device.
GPS Accuracy	Indicates the accuracy of the GPS location of the device rounded up to the nearest meter measurement. For example, if the accuracy is determined to be within 17.9 meters, 18 is returned in the reply. Returned only for mobile devices.
Device Latitude	Latitude of the GPS location of the device returned in the format degrees.minutes. For example: -37.82465426 Returned only for mobile devices.
Device Longitude	Longitude of the GPS location of the device returned in the format degrees.minutes. For example: 145.22554548 Returned only for mobile devices.
Jailbreak/Root Privileges	Indicates that a mobile device has root privileges. This form of privilege escalation is known as "jailbreaking" on iOS devices. This field returns a numerical value that indicates the number of root elements or "jailbreaks" detected on the device. 0 indicates that there are no root elements or jailbreaks detected. Returned only for mobile devices.
Jailbreak/Root Reason	Additional information that describes the elements on the device that triggered the escalation to root privileges or "jailbreak." See the field description for Jailbreak/Root Privileges . Returned only for mobile devices.
True IP Address	Customer IP address detected by the application.
True IP Address Activities	Actions associated with the true IP address.
True IP Address Attributes	Attributes associated with the true IP address.
Proxy IP Address	If applicable, IP address substituted for the true IP address.
Proxy IP Address Activities	Actions associated with the proxy IP address.
Proxy IP Address Attributes	Attributes associated with the proxy IP address.
Information Codes	Codes specific to the elements of the fingerprint.

Available Actions

Using the Available Actions menu in the Case Management Details window, you can add the fingerprint to your positive or negative list or remove it from history. If you choose Mark as Suspect, the Transaction Marking Tool window appears with all the data that you can add to the negative list for that order, including the fingerprint. The available data can differ from order to order. To add the fingerprint to your negative list, check the **Device Fingerprint** box in the Transaction Fields pane.

Similar Searches

Using the Similar Searches menu in the Case Management Details window, you can review other orders placed from the same computer or device by searching for orders that contain the same device fingerprint. The menu options appear only when the data is present in the order. In other words, you can search for other devices with the same fingerprint only when the current order contains a fingerprint. Smart ID can also be used to search when a Smart ID is present in the order.

The results table that is returned can contain up to 2,000 orders that correspond to your search parameters. To verify that you have the orders that you want, examine your search parameters, which are listed above the table. For example:

```
Results: Date: Aug 01 2019 12:00:00 AM - Feb 01 2014 06:55:49 PM |
Device Fingerprint 284928483475 | Transactions: 1568
```

Customer Lists

You can manually add device fingerprints to your positive or negative list from the List Addition window. (**Decision Manager > Lists > List Addition**)

You can also search customer lists for fingerprints. The fingerprint appears in downloaded reports.

Information Codes

You can view information codes in the AFS Information pane of the Case Management Details window.

API Fields and Information Codes

In addition to replacing the merchant and session IDs in your web page or your mobile application, you must send the session ID to CyberSource in your API request and be prepared to receive specific fields and information codes in the reply.

Simple Order API



Important

If you process call center orders, do not submit device fingerprint or IP address information in the requests of those orders because the device fingerprint or IP address information is for the call center and not for the customer who places the order.

Request Fields

Table 5 Simple Order API Request Fields

Field	Description	Used By: Required (R) or Optional (O)	Data Type & Length
deviceFingerprintHash	Field that contains the unique identifier of the device that is returned in the afsReply_deviceFingerprint_hash API reply field. To use this request field, you must use version 1.103 or later of the Simple Order API schema.	riskUpdate Service (O)	String (255)

Table 5 Simple Order API Request Fields (Continued)

Field	Description	Used By: Required (R) or Optional (O)	Data Type & Length
deviceFingerprintID	<p>Field that contains the session ID that you send to Decision Manager/FME to obtain the device fingerprint information. The string can contain uppercase and lowercase letters, digits, hyphen (-), and underscore (_). However, do not use the same uppercase and lowercase letters to indicate different session IDs.</p> <p>The session ID must be unique for each transaction and for each merchant ID. You can use any string that you are already generating, such as an order number or web session ID.</p> <p>The session ID must be unique for each page load, regardless of an individual's web session ID. If a user navigates to a profiled page and is assigned a web session, navigates away from the profiled page, then navigates back to the profiled page, the generated session ID should be different and unique. You may use a web session ID, but it is preferable to use an application GUID (Globally Unique Identifier). This measure ensures that a unique ID is generated every time the page is loaded, even if it is the same user reloading the page.</p> <p>To use this request field, you must use version 1.29 or later of the Simple Order API schema.</p>	Decision Manager (O)	String (88)
deviceFingerprintProxy IPAddress	IP address of the proxy if it is available.	riskUpdate Service (O)	String (15)
deviceFingerprintSmart ID	Field that contains the device identifier generated from attributes collected during profiling.	riskUpdate Service (O)	String (80)
deviceFingerprintTrue IPAddress	Customer's true IP address detected by the application.	riskUpdate Service (O)	String (15)

Reply Fields

All of these reply fields are returned by the Advanced Fraud Screen service (afsService). To receive these reply fields, you must use version 1.49 or later of the Simple Order API schema unless it is noted otherwise in the field description.

Table 6 Simple Order API Reply Fields

Field	Description	Data Type & Length
afsReply_deviceFingerprint_agentType	<p>Indicates whether a mobile device or a computer was used to initiate the session. If the session is initiated with a mobile device, this field indicates whether the mobile browser or mobile application is being used. This field can return the following values:</p> <ul style="list-style-type: none"> ■ <code>browser_computer</code>: Device is using a standard browser, which contains the fingerprinting tags. ■ <code>browser_mobile</code>: Device is using a mobile browser, which contains the fingerprinting tags. ■ <code>agent_mobile</code>: Device is using a mobile application, and fingerprinting mobile SDK tags are present in that mobile application. <p>To receive this reply field, you must use version 1.100 or later of the Simple Order API schema.</p>	String (255)
afsReply_deviceFingerprint_browserLanguage	<p>Comma-separated list of languages preferred or supported by the browser. When the browser supports more than one language, a Q value between 0 and 1 can be assigned to each language to indicate which language the browser prefers or supports. The preferred language is assigned the default value of 1, which may be omitted from the string.</p> <p>Examples:</p> <ul style="list-style-type: none"> ■ <code>en-us, en;q=0</code>: the browser prefers U.S. English but can support non-U.S. English. ■ <code>es, en-us; q=0.3, de;q=0.1</code>: the browser prefers Spanish (<code>es</code>) but can support U.S. English (<code>en-us; q=0.3</code>) and German (<code>de; q=0.1</code>). 	String (255)
afsReply_deviceFingerprint_cookiesEnabled	<p>Indicates whether cookies are enabled in the customer's browser. This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ <code>true</code> ■ <code>false</code> 	String (255)

Table 6 Simple Order API Reply Fields (Continued)

Field	Description	Data Type & Length
afsReply_deviceFingerprint_dateTime	<p>The arrival time of the first fingerprint attribute for this session, expressed in the following format:</p> <p>YYYY-MM-DDThh:mm:ssZ</p> <p>2016-08-11T22:47:57Z equals August 11, 2016, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.</p> <p>To receive this reply field, you must use version 1.100 or later of the Simple Order API schema.</p>	String (255)
afsReply_deviceFingerprint_deviceLatitude	<p>Returned for mobile devices only.</p> <p>Latitude of the GPS location of the device returned in the format degrees.minutes. For example:</p> <p>-37.82465426</p> <p>To receive this reply field, you must use version 1.100 or later of the Simple Order API schema.</p>	String (255)
afsReply_deviceFingerprint_deviceLongitude	<p>Returned for mobile devices only.</p> <p>Longitude of the GPS location of the device returned in the format degrees.minutes. For example:</p> <p>145.22554548</p> <p>To receive this reply field, you must use version 1.100 or later of the Simple Order API schema.</p>	String (255)
afsReply_deviceFingerprint_deviceMatch	<p>Indicates whether the device was encountered before and whether enough attributes were gathered to identify the device. This field can return the following values:</p> <ul style="list-style-type: none"> ■ Success: Device fingerprint was previously encountered. ■ New_Device: Device was not previously encountered. ■ Not_Enough_Attribs: Not enough attributes were gathered to identify whether the device was previously encountered. <p>To receive this reply field, you must use version 1.100 or later of the Simple Order API schema.</p>	String (255)
afsReply_deviceFingerprint_firstEncounter	<p>Date that the device was first encountered. This value is returned in the format:</p> <p>yyyy-mm-dd</p> <p>To receive this reply field, you must use version 1.100 or later of the Simple Order API schema.</p>	String (255)

Table 6 Simple Order API Reply Fields (Continued)

Field	Description	Data Type & Length
afsReply_deviceFingerprint_flashEnabled	Whether Flash is enabled in the customer's browser. This field can contain one of these values: <ul style="list-style-type: none"> ■ true ■ false 	String (255)
afsReply_deviceFingerprint_flashOS	Device operating system as reported by Flash. To receive this reply field, you must use version 1.100 or later of the Simple Order API schema. Note This field is not returned for iOS applications.	String (255)
afsReply_deviceFingerprint_flashVersion	The version of Flash installed on the device. To receive this reply field, you must use version 1.100 or later of the Simple Order API schema. Note This field is not returned for iOS applications.	String (255)
afsReply_deviceFingerprint_gpsAccuracy	Returned for mobile devices only. Indicates the accuracy of the GPS location of the device rounded up to the nearest meter measurement. For example, if the accuracy is determined to be within 17.9 meters, 18 is returned in the reply. To receive this reply field, you must use version 1.100 or later of the Simple Order API schema.	String (255)
afsReply_deviceFingerprint_hash	Unique identifier of the computer.	String (255)
afsReply_deviceFingerprint_imagesEnabled	Indicates whether images are enabled in the customer's browser. This field can contain one of these values: <ul style="list-style-type: none"> ■ true ■ false 	String (255)
afsReply_deviceFingerprint_javascriptEnabled	Indicates whether JavaScript is enabled in the customer's browser. This field can contain one of these values: <ul style="list-style-type: none"> ■ true ■ false 	String (255)
afsReply_deviceFingerprint_jbRoot	Returned for mobile devices only. Detects whether a mobile device running an application that contains device fingerprinting code has root privileges. This form of privilege escalation is known as "jailbreaking" on iOS devices. This field returns a numerical value that indicates the number of root elements or "jailbreaks" detected on the device. A "0" indicates that there are no root elements or jailbreaks detected. To receive this reply field, you must use version 1.100 or later of the Simple Order API schema.	Integer (255)

Table 6 Simple Order API Reply Fields (Continued)

Field	Description	Data Type & Length
afsReply_deviceFingerprint_ jbRootReason	Returned for mobile devices only. Returns additional information that describes the elements on the device that triggered the escalation to root privileges. To receive this reply field, you must use version 1.100 or later of the Simple Order API schema.	String (255)
afsReply_deviceFingerprint_ profileDuration	Total time in milliseconds to process the profiling request. To receive this reply field, you must use version 1.100 or later of the Simple Order API schema.	Integer (255)
afsReply_deviceFingerprint_ profiledURL	URL of the profiled page. To receive this reply field, you must use version 1.100 or later of the Simple Order API schema.	String (255)
afsReply_deviceFingerprint_ proxyIPAddress	IP address of the proxy if it is available.	String (255)

Table 6 Simple Order API Reply Fields (Continued)

Field	Description	Data Type & Length
afsReply_deviceFingerprint_proxyIPAddressActivities	<p>Actions associated with the proxy IP address. This field can contain one or more of these values, separated by carets (^):</p> <ul style="list-style-type: none"> ■ BANK: IP address belongs to a financial organization. ■ CLICK_FRAUD: IP address has been used for click fraud. ■ CONNECTING_TO_BOTNET: IP address has been connected to a botnet. ■ CONNECTING_TO_MALWARE_SITE: IP address has been connected to a malware site. ■ DNS_CONNECTION_ANOMALY: IP address has had DNS connection anomaly. ■ INSTANT_MSG: IP address has been used for instant messaging. ■ IRC_CONNECTION_ANOMALY: IP address has been connected to a suspicious IRC server. ■ LEGITIMATE: IP address has been legitimate. ■ MALWARE: IP address has been used for malware. ■ NIGERIAN: IP address has been used for Nigerian email or spam. ■ OTHER: IP has been involved in other activities. ■ P2P: IP address has been used for peer-to-peer communication. ■ PHISH: IP address has been used for phishing. ■ SPAM: IP address has been used to send spam. ■ TCP_SCAN_FLAG: IP address has been used as TCP port scanner. ■ UDP_SCAN_FLAG: IP address has been used as UDP port scanner. 	String (255)

Table 6 Simple Order API Reply Fields (Continued)

Field	Description	Data Type & Length
afsReply_deviceFingerprint_proxyIPAddressAttributes	<p>Characteristics associated with the proxy IP address. This field can contain one or more of these values, separated by carets (^):</p> <ul style="list-style-type: none"> ■ BOGON: IP address has been part of a range of bogus IP addresses. ■ BOTNET_ZOMBIE: IP address has been either a zombie or a botnet. ■ DYNAMIC: IP address has been dynamic. ■ HIJACKED: IP address has been part of a range of hijacked IP addresses. ■ NAME_SERVER: IP address has been a name server. ■ OPEN_PROXY: IP address has been an open proxy. ■ OPEN_RELAY: IP address has been an open relay. ■ PORTAL: IP address has been a portal. ■ PROXY: IP address has been a proxy. ■ RANGE: IP address has been part of a range of IP addresses. ■ STATIC: IP address has been static. 	String (255)
afsReply_deviceFingerprint_proxyServerType	<p>Type of proxy server based on the HTTP header. This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ Anonymous: presence of an HTTP header indicates the presence of a proxy but does not disclose the client IP address. ■ Hidden: absence of an HTTP header indicates the presence of a proxy attempting to hide its purpose. Often returned for compromised servers or botnets that are used as proxies. ■ Transparent: presence of an HTTP header indicates the presence of a proxy and discloses the client IP address. This value usually corresponds to a proxy that filters corporate or ISP content. This value is the safest. 	String (255)
afsReply_deviceFingerprint_screenResolution	Screen resolution of the device. The value is a number in the format nnnnXmmmm.	String (255)
afsReply_deviceFingerprint_smartID	Device identifier generated from attributes collected during profiling.	String (255)
afsReply_deviceFingerprint_smartIDConfidenceLevel	Probability that the Smart ID is correctly identifying a returning device. The value ranges from 0 to 100. A high number is more likely to represent a returning device than a new device similar to a previously identified device. As the confidence level decreases, the probability of false positives increases.	Integer (3)

Table 6 Simple Order API Reply Fields (Continued)

Field	Description	Data Type & Length
afsReply_deviceFingerprint_timeOnPage	Time period in milliseconds that the device profiling page displays on the browser before it closes or the user navigates away from the page. To receive this reply field, you must use version 1.100 or later of the Simple Order API schema.	Integer (255)
afsReply_deviceFingerprint_trueIPAddress	Customer's true IP address detected by the application.	String (255)
afsReply_deviceFingerprint_trueIPAddressActivities	<p>Actions associated with the true IP address. This field can contain one or more of these values, separated by carets (^):</p> <ul style="list-style-type: none"> ■ BANK: IP address belongs to a financial organization. ■ CLICK_FRAUD: IP address has been used for click fraud. ■ CONNECTING_TO_BOTNET: IP address has been connected to a botnet. ■ CONNECTING_TO_MALWARE_SITE: IP address has been connected to a malware site. ■ DNS_CONNECTION_ANOMALY: IP address has had a DNS connection anomaly. ■ INSTANT_MSG: IP address has been used for instant messaging. ■ IRC_CONNECTION_ANOMALY: IP address has been connected to a suspicious IRC server. ■ LEGITIMATE: IP address has been legitimate. ■ MALWARE: IP address has been used for malware. ■ NIGERIAN: IP address has been used for Nigerian email or spam. ■ OTHER: IP has been involved in other activities. ■ P2P: IP address has been used for peer-to-peer communication. ■ PHISH: IP address has been used for phishing. ■ SPAM: IP address has been used to send spam. ■ TCP_SCAN_FLAG: IP address has been used as TCP port scanner. ■ UDP_SCAN_FLAG: IP address has been used as UDP port scanner. 	String (255)

Table 6 Simple Order API Reply Fields (Continued)

Field	Description	Data Type & Length
afsReply_deviceFingerprint_trueIPAddressAttributes	<p>Characteristics associated with the true IP address. This field can contain one or more information codes, separated by carets (^). This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ BOGON: IP address has been part of a range of bogus IP addresses. ■ BOTNET_ZOMBIE: IP address has been either a zombie or a botnet. ■ DYNAMIC: IP address has been dynamic. ■ HIJACKED: IP address has been part of a range of hijacked IP addresses. ■ NAME_SERVER: IP address has been a name server. ■ OPEN_PROXY: IP address has been an open proxy. ■ OPEN_RELAY: IP address has been an open relay. ■ PORTAL: IP address has been a portal. ■ PROXY: IP address has been a proxy. ■ RANGE: IP address has been part of a range of IP addresses. ■ STATIC: IP address has been static. 	String (255)
afsReply_deviceFingerprint_trueIPAddressCity	City associated with the true IP address. If the data is available, the content of this field is more reliable than other city information in the order because any cloaking by the customer has been removed.	String (255)
afsReply_deviceFingerprint_trueIPAddressCountry	Country associated with the true IP address. If the data is available, the content of this field is more reliable than other country information in the order because any cloaking by the customer has been removed.	String (255)
afsReply_deviceFingerprint_trueIPAddressState	State associated with the true IP address. If the data is available, the content of this field is more reliable than other state information in the order because any cloaking by the customer has been removed.	String (255)
afsReply_identityInfoCode	Change in customer identity elements. This field can contain one or more codes, separated by carets (^), for example: MORPH-C^MORPH-B. For a list of values, see "Excessive Customer Identity Changes," page 62 .	String (255)
afsReply_suspiciousInfoCode	The customer provided potentially suspicious information. This field can contain one or more codes, separated by carets (^), for example: BAD-FP^MM-TZTLO. For a list of values, see "Suspicious Data Information Codes," page 60 .	String (255)
afsReply_velocityInfoCode	Customer has a high order velocity. This field can contain one or more codes, separated by carets (^), for example: VEL-S-TIP^VEL-I-TIP. For a list of values, see "Global Velocity," page 60 .	String (255)

Simple Order API Request and Reply Examples

These examples show only the minimum fields required in order to process the order.

Example 10 Simple Order API Request

```
billTo_<address_fields>=Customer's billing information
shipTo_<address_fields>=Customer's shipping information
card_<account_information>=Customer's account information
billTo_ipAddress=12.345.67.890
billTo_firstName=john
billTo_lastName=doe
billTo_email=jdoe@example.com
deviceFingerprintID=7685380BB8A476AB4C21FE705DC3AA66
afsService_run=true
purchaseTotals_currency=USD
item_0_unitPrice=1.00
```

Example 11 Simple Order API Reply

```
afsReply_suspiciousInfoCode=BAD-FP^INTL-BIN^MM-TZTLO^MUL-EM^RISK-DEV
afsReply_afsFactorCode=F
afsReply_afsResult=99
afsReply_hostSeverity=1
afsReply_identityInfoCode=MORPH-B^MORPH-C^MORPH-FB^MORPH-FE^MORPH-FP
afsReply_internetInfoCode=MM-IPBC
afsReply_ipCity=los angeles
afsReply_ipCountry=us
afsReply_ipRoutingMethod=standard
afsReply_ipState=ca
afsReply_reasonCode=481
afsReply_velocityInfoCode=VELS-FP
afsReply_deviceFingerprint_cookiesEnabled=true
afsReply_deviceFingerprint_flashEnabled=true
afsReply_deviceFingerprint_imagesEnabled=false
afsReply_deviceFingerprint_javascriptEnabled=true
afsReply_deviceFingerprint_trueIPAddress=66.185.179.2
afsReply_deviceFingerprint_smartID=278682734918374
afsReply_deviceFingerprint_smartIDConfidenceLevel=96
decision=REJECT
merchantReferenceCode=10679256010963322294714
purchaseTotals_currency=USD
reasonCode=481
```

SCMP API



Important

If you process call center orders, do not submit device fingerprint or IP address information in the requests of those orders because the device fingerprint or IP address information is for the call center and not for the customer who places the order.

Request Fields

Table 7 SCMP API Request Field

Field	Description	Used By: Required (R) or Optional (O)	Data Type & Length
device_fingerprint_hash	Field that contains the unique identifier of the device that is returned in the score_device_fingerprint_hash API reply field.	ics_risk_update (O)	String (255)
device_fingerprint_id	<p>Field that contains the session ID that you send to Decision Manager/FME to obtain the device fingerprint information. The string can contain uppercase and lowercase letters, digits, hyphen (-), and underscore (_). However, do not use the same uppercase and lowercase letters to indicate different session IDs.</p> <p>The session ID must be unique for each merchant ID. You can use any string that you are already generating, such as an order number or web session ID.</p> <p>The session ID must be unique for each page load, regardless of an individual's web session ID. If a user navigates to a profiled page and is assigned a web session, navigates away from the profiled page, then navigates back to the profiled page, the generated session ID should be different and unique. You may use a web session ID, but it is preferable to use an application GUID (Globally Unique Identifier). This measure ensures that a unique ID is generated every time the page is loaded, even if it is the same user reloading the page.</p>	Decision Manager (O)	String (88)
device_fingerprint_smart_id	Field that contains the device identifier generated from attributes collected during profiling.	ics_risk_update (O)	String (80)
proxy_ipaddress	IP address of the proxy if it is available.	ics_risk_update (O)	String (15)
true_ipaddress	Customer's true IP address detected by the application.	ics_risk_update (O)	String (15)

Reply Fields

These reply fields are all returned by the **ics_score** service.

Table 8 SCMP API Reply Fields

Field	Description	Data Type & Length
score_device_fingerprint_agent_type	<p>Indicates whether a mobile device or a computer was used to initiate the session. If the session is initiated with a mobile device, this field indicates whether the mobile browser or mobile application is being used. This field can return the following values:</p> <ul style="list-style-type: none"> ■ browser_computer: Device is using a standard browser that contains the fingerprinting tags. ■ browser_mobile: Device is using a mobile browser that contains the fingerprinting tags. ■ agent_mobile: Device is using a mobile application, and fingerprinting mobile SDK tags are present in that mobile application. 	String (255)
score_device_fingerprint_browser_language	<p>Comma-separated list of languages preferred or supported by the browser. When the browser supports more than one language, a Q value between 0 and 1 can be assigned to each language to indicate which language the browser prefers or supports. The preferred language is assigned the default value of 1, which may be omitted from the string.</p> <p>Examples:</p> <ul style="list-style-type: none"> ■ en-us, en;q=0: the browser prefers U.S. English but can support non-U.S. English. ■ es, en-us; q=0.3, de;q=0.1: the browser prefers Spanish (es) but can support U.S. English (en-us; q=0.3) and German (de; q=0.1). 	String (255)
score_device_fingerprint_cookies_enabled	<p>Indicates whether cookies are enabled in the customer's browser. This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ true ■ false 	String (255)
score_device_fingerprint_date_time	<p>The arrival time of the first fingerprint attribute for this session, expressed in the following format:</p> <p>YYYY-MM-DDThhmmssZ</p> <p>2016-08-11T22:47:57Z equals August 11, 2016, at 22:47:57 (10:47:57 p.m.). The T separates the date and the time. The Z indicates UTC.</p>	String (255)
score_device_fingerprint_device_latitude	<p>Returned for mobile devices only.</p> <p>Latitude of the GPS location of the device returned in the format degrees.minutes. For example:</p> <p>-37.82465426</p>	Decimal (255)

Table 8 SCMP API Reply Fields (Continued)

Field	Description	Data Type & Length
score_device_fingerprint_device_longitude	Returned for mobile devices only. Longitude of the GPS location of the device returned in the format degrees.minutes. For example: 145 . 22554548	Decimal (255)
score_device_fingerprint_device_match	Indicates whether the device was encountered before and whether enough attributes were gathered to identify the device. This field can return the following values: <ul style="list-style-type: none"> ■ Success: Device fingerprint was previously encountered. ■ New_Device: Device was not previously encountered. ■ Not_Enough_Attribs: Not enough attributes were gathered to identify whether the device was previously encountered. 	String (255)
score_device_fingerprint_first_encounter	Date that the device was first encountered. This value is returned in the format: yyyy-mm-dd	String (255)
score_device_fingerprint_flash_enabled	Whether Flash is enabled in the customer's browser. This field can contain one of these values: <ul style="list-style-type: none"> ■ true ■ false 	String (255)
score_device_fingerprint_flash_os	Device operating system as reported by Flash. Note This field is not returned for iOS applications.	String (255)
score_device_fingerprint_flash_version	The version of Flash installed on the device. Note This field is not returned for iOS applications.	String (255)
score_device_fingerprint_gps_accuracy	Returned for mobile devices only. Indicates the accuracy of the GPS location of the device rounded up to the nearest meter. For example, if the accuracy is determined to be within 17.9 meters, 18 is returned in the reply.	Decimal (255)
score_device_fingerprint_hash	Unique identifier of the computer.	String (255)
score_device_fingerprint_images_enabled	Indicates whether images are enabled in the customer's browser. This field can contain one of these values: <ul style="list-style-type: none"> ■ true ■ false 	String (255)
score_device_fingerprint_javascript_enabled	Whether JavaScript is enabled in the customer's browser. This field can contain one of these values: <ul style="list-style-type: none"> ■ true ■ false 	String (255)

Table 8 SCMP API Reply Fields (Continued)

Field	Description	Data Type & Length
score_device_fingerprint_jb_root	Returned for mobile devices only. Detects whether a mobile device running an application that contains device fingerprinting code has root privileges. This form of privilege escalation is known as “jailbreaking” on iOS devices. This field returns a numerical value that indicates the number of root elements or “jailbreaks” detected on the device. A “0” indicates that there are no root elements or jailbreaks detected.	Integer (255)
score_device_fingerprint_jb_root_reason	Returned for mobile devices only. Returns additional information that describes the elements on the device that triggered the escalation to root privileges.	String (255)
score_device_fingerprint_profile_duration	Total time in milliseconds to process the profiling request.	Integer (255)
score_device_fingerprint_profiled_url	URL of the profiled page. If the device fingerprinting mobile SDK is used, this reply field returns the custom URL that was specified in the <code>doProfileRequest()</code> function of your mobile application. See Step 3 of "Implementing the Device Fingerprinting SDK in Android Applications," page 19 , or Step 4 of "Implementing the Device Fingerprinting SDK in iOS Applications," page 24 .	String (255)
score_device_fingerprint_proxy_ipaddress	IP address of the proxy if it is available.	String (255)

Table 8 SCMP API Reply Fields (Continued)

Field	Description	Data Type & Length
score_device_fingerprint_proxy_ipaddress_activities	<p>Actions associated with the proxy IP address. This field can contain one or more of these values, separated by carets (^):</p> <ul style="list-style-type: none"> ■ BANK: IP address belongs to a financial organization. ■ CLICK_FRAUD: IP address has been used for click fraud. ■ CONNECTING_TO_BOTNET: IP address has been connected to a botnet. ■ CONNECTING_TO_MALWARE_SITE: IP address has been connected to a malware site. ■ DNS_CONNECTION_ANOMALY: IP address has had a DNS connection anomaly. ■ INSTANT_MSG: IP address has been used for instant messaging. ■ IRC_CONNECTION_ANOMALY: IP address has been connected to a suspicious IRC server. ■ LEGITIMATE: IP address has been legitimate. ■ MALWARE: IP address has been used for malware. ■ NIGERIAN: IP address has been used for Nigerian email or spam. ■ OTHER: IP has been involved in other activities. ■ P2P: IP address has been used for peer-to-peer communication. ■ PHISH: IP address has been used for phishing. ■ SPAM: IP address has been used to send spam. ■ TCP_SCAN_FLAG: IP address has been used as TCP port scanner. ■ UDP_SCAN_FLAG: IP address has been used as UDP port scanner. 	String (255)

Table 8 SCMP API Reply Fields (Continued)

Field	Description	Data Type & Length
score_device_fingerprint_proxy_ipaddress_attributes	<p>Characteristics of the proxy IP address. This field can contain one or more of these values, separated by carets (^):</p> <ul style="list-style-type: none"> ■ BOGON: IP address has been part of a range of bogus IP addresses. ■ BOTNET_ZOMBIE: IP address has been either a zombie or a botnet. ■ DYNAMIC: IP address has been dynamic. ■ HIJACKED: IP address has been part of a range of hijacked IP addresses. ■ NAME_SERVER: IP address has been a name server. ■ OPEN_PROXY: IP address has been an open proxy. ■ OPEN_RELAY: IP address has been an open relay. ■ PORTAL: IP address has been a portal. ■ PROXY: IP address has been a proxy. ■ RANGE: IP address has been part of a range of IP addresses. ■ STATIC: IP address has been static. 	String (255)
score_device_fingerprint_proxy_server_type	<p>Type of proxy server based on the HTTP header. This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ Anonymous: presence of an HTTP header indicates the presence of a proxy but does not disclose the client IP address. ■ Hidden: absence of an HTTP header indicates the presence of a proxy attempting to hide its purpose. Often returned for compromised servers or botnets that are used as proxies. ■ Transparent: presence of an HTTP header indicates the presence of a proxy and discloses the client IP address. This value usually corresponds to a proxy that filters corporate or ISP content. This value is the safest. 	String (255)
score_device_fingerprint_screen_resolution	Screen resolution of the device. The value is a number in the format nnnnXmmmm.	String (255)
score_device_fingerprint_smart_id	Device identifier generated from attributes collected during profiling.	String (255)
score_device_fingerprint_smart_id_confidence_level	Probability that the Smart ID is correctly identifying a returning device. The value ranges from 0 to 100. A high number is more likely to represent a returning device than a new device similar to a previously identified device. As the confidence level decreases, the likelihood of false positives increases.	Integer (3)

Table 8 SCMP API Reply Fields (Continued)

Field	Description	Data Type & Length
score_device_fingerprint_time_on_page	Time period in milliseconds that the device profiling page displays on the browser before it closes or the user navigates away from the page.	Integer (255)
score_device_fingerprint_true_ipaddress	Customer's true IP address detected by the application.	String (255)
score_device_fingerprint_true_ipaddress_activities	<p>Actions associated with the true IP address. This field can contain one or more of these values, separated by carets (^):</p> <ul style="list-style-type: none"> ■ BANK: IP address belongs to a financial organization. ■ CLICK_FRAUD: IP address has been used for click fraud. ■ CONNECTING_TO_BOTNET: IP address has been connected to a botnet. ■ CONNECTING_TO_MALWARE_SITE: IP address has been connected to a malware site. ■ DNS_CONNECTION_ANOMALY: IP address has had DNS connection anomaly. ■ INSTANT_MSG: IP address has been used for instant messaging. ■ IRC_CONNECTION_ANOMALY: IP address has been connected to a suspicious IRC server. ■ LEGITIMATE: IP address has been legitimate. ■ MALWARE: IP address has been used for malware. ■ NIGERIAN: IP address has been used for Nigerian email or spam. ■ OTHER: IP has been involved in other activities. ■ P2P: IP address has been used for peer-to-peer communication. ■ PHISH: IP address has been used for phishing. ■ SPAM: IP address has been used to send spam. ■ TCP_SCAN_FLAG: IP address has been used as TCP port scanner. ■ UDP_SCAN_FLAG: IP address has been used as UDP port scanner. 	String (255)

Table 8 SCMP API Reply Fields (Continued)

Field	Description	Data Type & Length
score_device_fingerprint_true_ipaddress_attributes	<p>Characteristics of the true IP address. This field can contain one or more information codes, separated by carets (^). This field can contain one of these values:</p> <ul style="list-style-type: none"> ■ BOGON: IP address has been part of a range of bogus IP addresses. ■ BOTNET_ZOMBIE: IP address has been either a zombie or a botnet. ■ DYNAMIC: IP address has been dynamic. ■ HIJACKED: IP address has been part of a range of hijacked IP addresses. ■ NAME_SERVER: IP address has been a name server. ■ OPEN_PROXY: IP address has been an open proxy. ■ OPEN_RELAY: IP address has been an open relay. ■ PORTAL: IP address has been a portal. ■ PROXY: IP address has been a proxy. ■ RANGE: IP address has been part of a range of IP addresses. ■ STATIC: IP address has been static. 	String (255)
score_device_fingerprint_true_ipaddress_city	City associated with the true IP address. If the data is available, the content of this field is more reliable than other city information in the order because any cloaking by the customer has been removed.	String (255)
score_device_fingerprint_true_ipaddress_country	Country associated with the true IP address. If the data is available, the content of this field is more reliable than other country information in the order because any cloaking by the customer has been removed.	String (255)
score_device_fingerprint_true_ipaddress_state	State associated with the true IP address. If the data is available, the content of this field is more reliable than other state information in the order because any cloaking by the customer has been removed.	String (255)
score_identity_info	Change in customer identity elements, such as address or account number. This field can contain one or more codes, separated by carets (^), for example: MORPH-C^MORPH-B. For a list of values, see "Information Codes," page 60 .	String (255)
score_suspicious_info	The customer provided potentially suspicious information. This field can contain one or more codes, separated by carets (^), for example: BAD-FP^MM-TZTLO. For a list of values, see "Suspicious Data Information Codes," page 60 .	String (255)
score_velocity_info	Customer has a high order velocity. This field can contain one or more codes, separated by carets (^), for example: VEL-S-TIP^VELI-TIP. For a list of values, see "Global Velocity," page 60 .	String (255)

SCMP API Request and Reply Examples

These examples show only the minimum fields required to process the order.

Example 12 SCMP API Request

```
bill_<address_fields>=Customer's billing address
ship_to_<address_fields>=Customer's shipping address
customer_<account_information>=Customer's account information
customer_ipaddress=12.345.67.890
customer_firstname=john
customer_lastname=doe
customer_email=jdoe@example.com
device_fingerprint_id=7685380BB8A476AB4C21FE705DC3AA66
ics_applications=ics_score
currency=USD
merchant_ref_number=10679256010963322294714
offer0=amount:1.00
```

Example 13 SCMP API Reply

```
score_address_info=COR-BA^MM-A^MM-C^MM-ST^MM-Z^UNV-ADDR
score_suspicious_info=BAD-FP^INTL-BIN^MM-TZTLO^MUL-EM^NON-LN^RISK-DEV
score_factors=Y
score_host_severity=1
score_identity_info=MORPH-B^MORPH-C^MORPH-FB^MORPH-FE^MORPH-FP
score_internet_info=MM-IPBC
score_ip_city=los angeles
score_ip_country=us
score_ip_routing_method=standard
score_ip_state=ca
score_device_fingerprint_cookies_enabled=true
score_device_fingerprint_flash_enabled=true
score_device_fingerprint_images_enabled=false
score_device_fingerprint_javascript_enabled=true
score_device_fingerprint_true_ipaddress=66.185.179.2
score_device_fingerprint_smart_id=278682734918374
score_device_fingerprint_smart_id_confidence_level=96
score_rcode=0
score_rflag=REJECT
score_rmsg=...reject...
score_score_result=99
score_velocity_info=VELS-FP
```

Information Codes

Global Velocity

Table 9 Global Velocity Codes

Code	Description
VELS-FP	The device fingerprint has been used several times during the short interval.
VELI-FP	The device fingerprint has been used several times during the medium interval.
VELL-FP	The device fingerprint has been used several times during the long interval.
VELV-FP	The device fingerprint has been used several times during the very long interval.
VELS-TIP	The true IP address has been used several times during the short interval.
VELI-TIP	The true IP address has been used several times during the medium interval.
VELL-TIP	The true IP address has been used several times during the long interval.

Suspicious Data Information Codes

Table 10 Suspicious Data Information Codes

Code	Description
ANOM-BLANG	The browser string contains unusual words or patterns.
ANOM-BSTR	The browser string contains unexpected information.
ANOM-FLASH	Flash is installed but not enabled.
ANOM-IMAGE	An anomaly was detected that is associated with images loading in the browser.
ANOM-LANG	An anomaly was detected that is associated with the browser's language setting.
ANOM-OS	The operating system indicated by the browser is inconsistent with the operating system that is detected with other system checks.
ANOM-SESS	An unexpected change occurred in the session.
ANOM-SRAT	The screen aspect ratio is outside the expected ranges.
ANOM-SRES	The screen resolution is outside the expected ranges.
ANOM-TZO	The time zone offset is inconsistent with the operating system.
BAD-FP	The device is risky.
DEV-MOB	The Smart ID detected a mobile device.
MASK-FP	The device history is masked.
MM-TZTLO	The device's time zone is inconsistent with the country's time zones.
NEW-FP	The Smart ID detected a new device.
RISK-DEV	Some of the device characteristics are risky.

Table 10 Suspicious Data Information Codes (Continued)

Code	Description
RISK-PIP	The proxy IP address is risky. It was recently used as botnet or for spam or hacking purposes.
RISK-TIP	The true IP address is risky. It was recently used as botnet or for spam or hacking purposes.

Excessive Digital Identity Changes

Table 11 Excessive Digital Identity Changes

Code	Description
MORPH-FB	The device fingerprint has occurred several times with multiple billing addresses.
MORPH-FC	The device fingerprint has occurred several times with multiple account numbers.
MORPH-FE	The device fingerprint has occurred several times with multiple email addresses.
MORPH-FI	The device fingerprint has occurred several times with multiple IP addresses.
MORPH-FP	The device fingerprint has occurred several times with multiple phone numbers.
MORPH-FPIP	The device fingerprint has occurred several times with multiple proxy IP addresses.
MORPH-FPLO	The device fingerprint has occurred several times in multiple proxy IP address locations.
MORPH-FRES	The device fingerprint has occurred several times with multiple screen resolutions.
MORPH-FS	The device fingerprint has occurred several times with multiple shipping addresses.
MORPH-FTIP	The device fingerprint has occurred several times with multiple true IP addresses.
MORPH-FTLO	The device fingerprint has been used several times in multiple true IP address locations.
MORPH-FTZ	The device fingerprint has occurred several times in multiple time zones.
MORPH-TF	The true IP address has occurred several times with multiple devices.
MORPH-TPIP	The true IP address has occurred several times with multiple proxy IP addresses.
MORPH-TPLO	The true IP address has occurred several times in multiple proxy IP address locations.
MORPH-TRES	The true IP address has occurred several times with multiple screen resolutions.
MORPH-TTZ	The true IP address has occurred several times in multiple time zones.

Excessive Customer Identity Changes

You receive an information code when more than two identity changes occur for one customer. *Customer identity* refers to one or more of these elements: account and phone numbers, billing, shipping, fingerprint, email, and IP addresses.

Table 12 Excessive Customer Identity Changes

Code	Description
MORPH-B	The billing address has been used several times with multiple customer identities.
MORPH-C	The account number has been used several times with multiple customer identities.
MORPH-E	The email address has been used several times with multiple customer identities.
MORPH-I	The IP address has been used several times with multiple customer identities.
MORPH-P	The phone number has been used several times with multiple customer identities.
MORPH-S	The shipping address has been used several times with multiple customer identities.

Code Examples for Mobile Implementation

This appendix includes code examples for Android and iOS mobile implementations.

Android Code Example

The following excerpt from an Android application shows how to set the `doProfileRequest()` function calling options.

Example 14 Android Code

```
@Override
protected void onCreate(Bundle savedInstanceState)
{
    Log.d(TAG, "onCreate");
    super.onCreate(savedInstanceState);
    setContentView(R.layout.login_activity);

    /*
     * When the device rotates the activity will be destroyed and created again, although
     * before destroying the state of the app will be written in a Bundle. To avoid
     * unnecessary profiling (i.e. when device rotates) we check the bundle here and skip
     * profiling if the device is rotated.
     */
    if(savedInstanceState == null)
    {
        //Resetting the status of the application to make sure we have a clean activity
        reset();

        /*
         * Creating a config instance to be passed to the init() method. Note this
         * instance must include orgId and application context otherwise the init()
         * method will fail.
         */
        Config config = new Config().setOrgId(ORG_ID           // (REQUIRED) Organisation ID
                                   .setFPServer(FP_SERVER)    // (REQUIRED) Enhanced
                                                           fingerprint server
                                   .setContext(getApplicationContext())
                                                           // (REQUIRED) Application Context
    }
```

```

        .setTimeout(10, TimeUnit.SECONDS)
            // (OPTIONAL) Set the connection
            time out in seconds
        .setRegisterForLocationServices(true)
            // (OPTIONAL) init() should request
            location updates
        .setRegisterForPush(true);
            // (OPTIONAL) enable Strong
            authentication notification
    /*
    * Call init to get some of the slow start up items completed before
    * we request a profile. This is a mandatory call, and requires, at a minimum,
    * the application context and the orgId.
    *
    * Only the first call to init() will use the configuration object, Subsequent
    * calls will be ignored. init() can fail due to illegal argument or state in
    * which cases throws exception to draw attention to the programming problem.
    * Failure cases include malformed organisation id, malformed fingerprint
    * server, missing okhttp library without setting .setDisableOKHTTP(true), etc
    */
    TrustDefender.getInstance().init(config);

    //Init was successful or there is a valid instance to be used for further calls.
    //Fire a profile request
    Log.e(TAG, "Successfully init-ed ");
    doProfile();
}
}
void doProfile()
{
    // (OPTIONAL) Retrieve the version of the SDK
    Log.i(TAG, "Using: " + TrustDefender.version);

    // (OPTIONAL) Assign some custom attributes to be included with the profiling
    information
    List<String> list = new ArrayList<String>();
    list.add("attribute 1");
    list.add("attribute 2");

    ProfilingOptions options = new ProfilingOptions().setCustomAttributes(list);

    // Fire off the profiling request. We could use a more complex request,
    // but the minimum works fine for our purposes.
    Profile.Handle profilingHandle =
    TrustDefender.getInstance().doProfileRequest(options,
        new CompletionNotifier()); // (REQUIRED) The end notifier must be passed to
    the doProfileRequest() method

    // Session id can be collected here
    Log.d(TAG, "Session id = " + profilingHandle.getSessionID());

    /*
    * profilingHandle can also be used to cancel this profile if needed

```



```

        *
        * profilingHandle.cancel();
        * */
    }
    /**
     * Used for notification from the SDK. Any code that needs to be run upon profiling
     * completion should be called from here.
     * <p>
     * Note: Be careful about calling UI update functions from here, as any callbacks will
     * not happen from the UI thread.
     * </p>
     */
    private class CompletionNotifier implements EndNotifier
    {
        /**
         * This gets called when the profiling has finished.
         * We have to be careful here because we are not going to be called on the UI thread,
         * and if we want to update UI elements we can only do it from the UI thread.
         */
        @Override
        public void complete(Profile.Result result)
        {
            //Get the session id to use in API call (AKA session query)
            m_sessionID = result.getSessionID();

            Log.i("<YourAppName>Completion", "Profile completed with: " +
result.getStatus().toString()
                + " - " + result.getStatus().getDesc());

            /**
             * Profiling is complete, so login can proceed when the Login button is clicked.
             */
            setProfileFinished(true);
            /**
             * Fire off a package scan. This will run in the background and process any newly
             * installed apps
             * We pass a value of 0 to disable the timeout; it will run until all
             * packages are scanned.
             * PackageScan runs on a different thread and doesn't interfere with
             * <YourAppName> app or profiling request
             */
            TrustDefender.getInstance().doPackageScan();
            /**
             * The Login button is clicked before the profiling is finished, therefore we
             * should login
             */
            if(isLoginClicked())
            {
                login();
            }
        }
    }
}

```

iOS Code Example

The following excerpt from an iOS application shows how to set the `doProfileRequest()` function calling options where `ApplicationName` is the name of your iOS application:

Example 15 iOS Code

```
- (instancetype) init
{
    self = [super init];

    _sessionID = nil;
    _timeout    = @10;
    _profile    = [THMTrustDefender sharedInstance];

    static dispatch_once_t configureOnce = 0;

    // The [_profile configure] method is effective only once and subsequent calls to it
    // will be ignored. By having a dispatch_once here we make sure configure is called
    // only once, although using a dispatch_once is not technically required.
    dispatch_once(&configureOnce,
        ^{
            // The profile.configure method is effective only once and subsequent
            // calls to it will be ignored.
            // Note that configure may throw NSError if NSDictionary key/
            // value(s) are invalid.
            // This only happen due to programming error, therefore we don't
            // catch the exception to make sure there is no error in our
            // configuration dictionary
            [_profile configure:@{
                THMOrgID : ORG_ID,
                // (REQUIRED) Enhanced fingerprint server
                THMFingerprintServer : FP_SERVER,
                // (OPTIONAL) Set the connection
                // timeout, in seconds
                THMTimeout : _timeout,
                // (OPTIONAL) If Keychain Access sharing groups
                // are used, specify like this
                THMKeychainAccessGroup:
                @"TEAMID.com.threatmetrix",
                // (OPTIONAL) Register for location service
                // updates
                // Note that this call can prompt the user for
                // permissions. The related call
                // registerLocationServices will only activate
                // location services have already been granted.
                // But in this case, we want the request to happen
                THMLocationServicesWithPrompt: @YES,
                //(OPTIONAL) Register for ThreatMetrix Strong
```

```

        // Authentication, using this feature needs some
        // setup in the AppDelegate class.
        THMRegisterForPush: @YES,
    }];

    });

    return self;
}

-(void)doProfile
{
    // (OPTIONAL) Retrieve the version of the mobile SDK
    NSLog(@"Using: %@", self.profile.version);

    NSArray *customAttributes = @[@"attribute 1", @"attribute 2"];

    // Fire off the profiling request.
    THMProfileHandle *profileHandle = [self.profile
        doProfileRequestWithOptions:@{
            // (OPTIONAL) Assign some custom
            // attributes to be included with the
            // profiling information
            THMCustomAttributes: customAttributes
        }
        andCallbackBlock:^(NSDictionary *result)
        {
            THMStatusCode statusCode = [[result
valueForKey:THMProfileStatus] integerValue];
            // If we registered a delegate, this function
            // will be called once the profiling is
            // complete
            if(statusCode == THMStatusCodeOk)
            {
                // No errors, profiling succeeded!
            }

            NSLog(@"Profile completed with: %s and session ID: %@", statusCode ==
THMStatusCodeOk ? "OK"
                : statusCode == THMStatusCodeNetworkTimeoutError ? "Timed out"
                : statusCode == THMStatusCodeConnectionError ? "Connection Error"
                : statusCode == THMStatusCodeHostNotFoundError ? "Host not found error"
                : statusCode == THMStatusCodeInternalError ? "Internal Error"
                : statusCode == THMStatusCodeInterruptedError ? "Interrupted"
                : "other",
                [result valueForKey:THMSessionID]);
        }];

    // Session id can be collected here (to use in API call (AKA session query))
    self.sessionID = profileHandle.sessionID;
    NSLog(@"Session id is %@", self.sessionID);

    /*
    * profileHandle can also be used to cancel this profile if needed
    *
    * [profileHandle cancel];
    */
}

```

```

    * */
}

```

Example 16 iOS Code (Swift v4.0)

```

class <YourAppName>ProfileController: NSObject
{
    var profile: THMTrustDefender!
    dynamic var loginOkay: Bool = false
    var sessionID: String = ""

    override init()
    {
        super.init()

        //Get a singleton instance of TrustDefenderMobile
        profile = THMTrustDefender.sharedInstance() as THMTrustDefender

        //Configuration only fails due to programming error, therefore by using an assert
        //here we make sure there is no error in our configuration object
        assert(profile.configure([
            THMOrgID:<ORG_ID>,
            // (OPTIONAL) Set the connection timeout, in
            // seconds
            THMTimeout : 10,
            // (OPTIONAL) If Keychain Access sharing groups
            // are used, specify like this
            THMKeychainAccessGroup: "TEAMID.com.threatmetrix",
            //(OPTIONAL) Register for location service updates
            // Note that this call can prompt the user for
            // permissions. The related call
            // registerLocationServices will only activate
            // location services have already been
            // granted. But in this case, we want the
            // request to happen
            THMLocationServicesWithPrompt: true,

            ]),
            "Configuration failed")
        }

    func doProfile()
    {
        // (OPTIONAL) Retrieve the version of the mobile SDK
        NSLog("Using \(profile.version)")

        let customAttributes : [String : Array<String>] = [THMCustomAttributes:
["attribute 1", "attribute 2"]]

        loginOkay = false
    }
}

```

```

        // Fire off the profiling request.
        let status: THMStatusCode = profile.doProfileRequest(options: customAttributes,
andCallbackBlock:
            {(result: [AnyHashable : Any]?) -> Void in

                let results:NSDictionary! = result! as NSDictionary
                let status:THMStatusCode =
THMStatusCode(rawValue:(results.value(forKey: THMProfileStatus) as!
NSNumber).intValue)!

                self.sessionID = results.value(forKey: THMSessionID) as! String
                if(status == .ok)
                {
                    // No errors, profiling succeeded!
                }

                NSLog("Profile completed with: %@ and session ID: \(self.sessionID)",
                    status == .ok                ? "OK" :
                    status == .networkTimeoutError ? "Timed out" :
                    status == .connectionError      ? "Connection Error" :
                    status == .hostNotFoundError    ? "Host Not Found Error" :
                    status == .internalError         ? "Internal Error" :
                    status == .interruptedError      ? "Interrupted Error" :
                    "Other"
                )
                self.loginOkay = true
            })

        if(status == .ok)
        {
            // The profiling successfully started
            NSLog("Profiling started successfully")
        }
        else
        {
            // We errored out, allow the login to proceed
            // The most common case of this error is THM_NotYet which means another
            // profiling is in progress and we should wait for that one to finish.
            loginOkay = true
        }
    }
}

```

Device Fingerprinting Cookie FAQ

Because of developing regulations regarding cookie usage in the European Union,¹ CyberSource has received questions about how its services use cookies. This information is included here because CyberSource Decision Manager Device Fingerprinting and CyberSource Decision Manager Account Takeover Protection Service use cookies.

1 What is a cookie?

A cookie is a small file, typically consisting of letters and numbers, which is downloaded to and stored on a user's computer or other electronic device when the user visits certain web sites. Information from cookies is used for a variety of purposes. For example, cookies can be used to enhance security or configure a web site to make it more convenient for a visitor.

2 Does CyberSource Decision Manager set cookies on users' computers?

Yes, but only if you are using device fingerprinting or the Decision Manager Account Takeover Protection Service. If you are not using device fingerprinting or the Decision Manager Account Takeover Protection Service, Decision Manager does not set any cookies.

3 What purpose does the cookie serve? Will the service function without the cookie?

If you are using device fingerprinting or the Decision Manager Account Takeover Protection Service, one cookie is dropped as described in the following chart:

Purpose	Data Stored	Will the service function without the cookie?	Persistent?
Provides identification of a returning device.	The user's device fingerprint generated by CyberSource's device fingerprint technology vendor.	Decision Manager will function without the cookie.	Yes, for five years.

¹ This information is not intended to be legal advice. CyberSource recommends that you seek advice from independent counsel regarding your obligations regarding the use of cookies under applicable law.

4 Does CyberSource obtain user consent for this cookie?

No. CyberSource is a third-party vendor and does not have contact or a direct relationship with your users. Under your agreement with CyberSource, it is the merchant's responsibility to provide their users any legally required notices or obtain necessary consent in order to set cookies.

Please contact us if you have any questions.