# POODLE SSL 3.0 Vulnerability

Merchant Response Guide

## Overview

### What is POODLE?

POODLE is an internet security vulnerability that impacts the Secure Sockets Layer (SSL) 3.0 protocol, which was designed to ensure secure connections when surfing on the Internet. When exploited, this vulnerability enables a cyber criminal to gain access to connections considered secure via this widespread (but 15-year-old) security protocol.

### How is PayPal responding?

PayPal will completely disable SSL 3.0 support in a timeframe to be announced via PayPal Notify; however, based on security monitoring, we may need to move quickly to protect our customers so time is of the essence in making changes. Unfortunately, we realize shutting off SSL 3.0 may cause compatibility problems for a few of our customers resulting in the inability to pay with PayPal on some merchant sites or other processing issues that we are still identifying. To enable your assessment and potential remediation, we've put together this Merchant Response Guide to ensure your integration is secure from this vulnerability.

## What you need to do…

### 1. Test your current integration against the PayPal Sandbox

If you are directly integrated with PayPal, follow the steps below:

**NOTE:** If you are integrated through a Partner, no further action is required on your part. We are working with our Partners to resolve the SSL 3.0 issue.

1. Point your test environment to our Sandbox: https://developer.paypal.com/docs/classic/lifecycle/ug_sandbox/
   - SSL 3.0 has already been disabled on the PayPal Sandbox, so if you can successfully make an application programming interface (API) request you are not using SSL 3.0.
2. If your request fails, check your logs to see why.
   - If you see an error similar to those shown below, then you are using SSL 3.0 and will need to configure your secure connection to use Transport Layer Security (TLS).

```
* Unknown SSL protocol error in connection to api-3t.sandbox.paypal.com:-9824
```

OR

```
140062736746144:error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version
number:s3_pkt.c:337:
...
New, (NONE), Cipher is (NONE)
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
SSL-Session:
Protocol: SSLv3
...
```

## 2. Update to TLS

All PayPal customers are required to disable SSL 3.0 for client interactions as soon as possible and upgrade to TLS. The following table provides basic guidelines on how to update to TLS using common languages and connection methods. Your exact settings may vary…

| Connection Method | Action | |
|---|---|---|
| **PayPal SDK** | No current PayPal Software Development Kit (SDK) versions or languages use SSL 3.0. However, since the Java and PHP SDKs were recently updated to address this issue, all merchants using these SDKs (or legacy SDKs) will need to update to the latest version.<br>• For information on the latest SDK versions, see: http://paypal.github.io/sdk/ | |
| **API Endpoint** | Ensure you are connecting to PayPal endpoints using TLS 1.0 or 1.2 (not all API endpoints currently support TLS 1.1). See the table below to set the TLS protocol for the language you are using. | |
| | **Language** | **Action** |
| | **Ruby** | Set the TLS protocol in the OpenSSL::SSL::SSLContext.<br>• For more details, see: http://ruby-doc.org/stdlib-1.9.3/libdoc/openssl/rdoc/OpenSSL/SSL/SSLContext.html |
| | **Python** | Set the TLS protocol in the ssl.SSLContext.<br>• For more details, see: https://docs.python.org/2/library/ssl.html#ssl.SSLContext |
| | **Node.js** | Use the correct renegotiation limit as specified here:<br>• http://nodejs.org/api/tls.html#tls_client_initiated_renegotiation_attack_mitigation |
| | **PHP** | Set CURLOPT_SSLVERSION to CURL_SSLVERSION_TLSv1 in your Curl options.<br>• For more details, see: http://curl.haxx.se/libcurl/c/CURLOPT_SSLVERSION.html |
| | **Java** | Set the TLS protocol in the javax.net.ssl.SSLContext.<br>• For more details, see: http://docs.oracle.com/javase/7/docs/technotes/guides/security/jsse/JSSERefGuide.html |
| | **C#** | Use SecurityProtocolType Tls.<br>• For more details, see: http://msdn.microsoft.com/en-us/library/system.net.securityprotocoltype%28v=vs.110%29.aspx |

## 3. Issue new credentials (optional)

After you've successfully tested and upgraded to TLS, you may want to reissue and download new API credentials for any PayPal API requests. This step is recommended, but not required. Please make a risk-based decision for your business and customers.

- If you are using **Certificate** authentication, no action is required.
- If you are using **Signature** authentication, see: https://developer.paypal.com/docs/classic/api/apiCredentials/
- If you are using **OAuth** authentication, see: https://developer.paypal.com/docs/integration/admin/manage-apps/

# Thank You

Thank you for your prompt attention to this issue and understanding of our approach. Though we recognize this necessary step may cause compatibility issues, we can't stress enough that this short-term inconvenience is heavily outweighed by our joint promise to our respective customers that we will keep their financial details safe. We plan to keep our customers up to date on how we are addressing this issue via the appropriate channels, including PayPal Forward, our Twitter handle, Customer Service and for merchants, through our Merchant Services team. We appreciate your patience and understanding as we work around the clock to better serve you and keep you safe.