Authorize.NET Technical Updates

A White Paper

06/03/2016

eGrove Systems Corporation

Contents

Authorize.NET Technical Updates

Introduction

Authorize.Net is a payment gateway service provider allowing merchants to accept credit card and electronic check payments through their Web site and over an IP (Internet Protocol) connection.

In September 2004, Authorize.Net's servers were hit by a Distributed Denial of Service (DDoS) attack. The DDoS attack lasted for over one week and caused a virtual shut down of the payment gateway's service. The attackers demanded money from Authorize.net in exchange for stopping the attack.

On July 2, 2009, 11pm, the entire web infrastructure for Authorize.net (main website, merchant gateway website, etc.) went offline and stayed down all morning July 3, 2009. None of the over 200,000 merchants who use Authorize.net payment gateway were able to process credit cards.

Problem Statement

Over the next couple of months, Authorize.Net is making several updates to our systems that you need to be aware of. They are all technical in nature and may require the assistance of your web developer or shopping cart/payment solution provider.

The following is a list of authorize.Net technical updates,

- Akamai SureRoute.
- Transaction and Batch Ids.
- RC4 Cipher Disablement.
- TLS Remediation for PCI DSS Compliance.

Proposed Solution(s)

Following are the proposed solutions for each technical updates from Authorize.NET.

Akamai SureRoute:

Akamai is a third-party cloud network service that routes and delivers Internet traffic. They provide a distributed network of servers and over 100,000 dynamic IP addresses, any of which could be used at any time.

What this means is that Authorize.net are changing how they route requests to the gateway in order to process. Rather than submitting the request directly to the system, they are routing them through a cloud service called Akamai. The reason for doing that is that it creates multiple routes to the Gateway, rather than relying on a single route, so if one of those routes has technical issues, the requests can be re-routed and still reach the gateway, which should improve the uptime of the Gateway.

Although Authorize.net have created some new URLs which can use the Akamai SureRoute system now, the existing URLs that MemberGate submits requests too, will automatically be updated in June 2016 to take advantage of the new service, so no update should be required to the code. On June 30th 2016, Authorize.net will update the existing URLs to go through the Akamai

system, not deprecate them. Therefore, no action is needed. You can change your URLs now to take advantage of the better reliability of the Akamai service but it is not in any way necessary. Here is how you make the change, which can be done today.

1. Log into the Administration Console of your Magento site.
2. Navigate to the System Configuration page.
3. Navigate down to the Payment Methods section on the left. It is far down the list so don't worry if you don't see it right away.
4. Expand the Authorize.net section of configuration and find the setting for "Gateway URL." This is what we will need to change.
5. Change the setting based on the table below, save and relax! You are all ready for the Authorize.net changes in 2016.

Transaction and Batch IDs:

In the coming months, due to system updates, it will be possible to receive Authorize.Net IDs (Transaction ID, Batch ID, etc.) that are not in sequential order.

For example, currently, if you receive a Transaction ID of "1000," you could expect that the next Transaction ID would not be less than 1000. However, after the updates, it will be possible to receive a Transaction ID less than the one previously received.

If your system contains any functionality that accepts Authorize.Net generated IDs to be sequential, please update it immediately so that you will not see any disruptions.

Additionally, please make sure that your solution does not restrict any Authorize.Net ID field to 10 characters. If you are required to define a character limit when storing any of our IDs, the limit should be no less than 20 characters.

RC4 Cipher Disablement:

We hear a lot about Cyber Security in the news these days. From HeartBleed to LogJam, vulnerabilities are being found every day and a need to fix them keeps arising. To this end, Authorize.net (and every other sane company out there) is removing support for a long insecure cipher suite from their servers. This suite is RC4. This only affects people who built systems that specify the RC4 cipher suite specifically, which would likely be very old and very insecure systems.

To explain it better, secure communications take place between two systems using very complex and advanced mathematics. Each year new and better systems are developed. As well, every year older, less secure systems are deprecated. When you visit a secure website with your phone or computer the first thing that normally happens is the two systems negotiate on which encryption technologies to use. Think of it like two strangers meeting in a desert. Neither knows what language the other speaks, so they each write out the name of the languages they speak in that language on a piece of parchment. Then they

swap those and read. If one looks down and can't read a line, clearly he doesn't speak that language. So the strangers choose the language they prefer and communication starts.

With that in mind, you should know the list that is exchanged between machines is very long and can include some very nasty old options. This wasn't seen as an issue for a long time because browsers and servers were smart and would decide to use the best option available. However, in the last couple years malicious folks have figured out techniques to trick browsers into asking for much weaker 'languages' which make it easier for those people to learn what is being communicated. To prevent this, Authorize.net is completely removing the option of using RC4 as a cipher suite. This was done by the Firefox team in September of 2015. Google Chrome also removed RC4 in January 2016. Microsoft still has yet to remove support for RC4 which is unfortunate.

TLS Remediation for PCI DSS Compliance:

As you may already be aware, new PCI DSS requirements state that all payment systems must disable TLS 1.0 by 2018. Though it is still in finalizing stages of plans for remediating TLS 1.0 in both sandbox and production, TLS 1.0 will be disabled in sandbox and production in early 2017. This is to ensure that it is compliant ahead of the PCI date.

In addition, discussion regarding the possibility of disabling TLS 1.1 is happening at the same time, because while it is not expressly forbidden, there are enough concerns surrounding it. TLS 1.2 is currently the strongest available protocol, and we strongly urge all merchants and developer partners to use it for their API. The following operating systems, components, and frameworks are known to support TLS 1.2:

Windows Server:	Version 2008 R2 and later. (Source)
.NET:	Version 4.5 and later. Requires Windows Server 2008 R2 SP1.(Source 1, Source 2)
OpenSSL:	Version 1.0.1 and later. (Source)
cURL:	Version 7.34.0 and later. (Source)
PHP:	Version 5.6 and later. Requires OpenSSL 1.0.1 and later. (Source)
Java:	JRE 1.7 / JDK 7 and later. (Source)
ColdFusion:	Version 10 with JRE 1.8; Version 11 with JRE 1.7 or greater.(Source)
Perl:	Depends on implementation. Net::SSLeay requires OpenSSL 1.0.1 and later.(Source)
Nginx:	Version 0.7.65/0.8.19 and later. Requires OpenSSL 1.0.1 and later.(Source 1, Source 2)
MacOS:	Version 10.9 AKA Mavericks.(Source)
iOS:	Version 5 and later.(Source)

Android OS:	Version 4.2 and later. Requires OpenSSL 1.0.1 and later (bundled by default).(Source)

Summary

The summary of the Authorize.Net technical update is given below and it is advised to make the updates with the assistance of either the web developer or the shopping cart provider.

1) Authorize.Net is now using **Akamai SureRoute** to optimize our Internet traffic routing, which includes your transaction requests. Akamai helps safeguard against interruptions caused by issues beyond Authorize.Net's direct control, such as Internet congestion, fiber cable cuts and other similar issues. Using Akamai is currently optional, but will be mandatory starting June 30th when we direct our existing transaction URLs on our end to connect through Akamai SureRoute. Upgrade your website or payment solution today, however, to take immediate advantage of Akamai's benefits.

2) It was previously announced about disabling the **RC4 cipher** suite in the production environment on May 31, 2016. Unfortunately, that date has been delayed. The new date is June 13th. However, RC4 has now been disabled in the sandbox, so you can test your system ahead of time. If you have a solution that relies on RC4 to communicate with our servers, please update it to a current, high-security cipher as soon as possible.

3) New **PCI DSS requirements** state that all payment systems must disable TLS 1.0 by 2018. Though it is still in finalizing stages of plans for remediating TLS 1.0 in both sandbox and production, TLS 1.0 will be disabled in sandbox and production in early 2017

4) Due to system updates, it will be possible to receive **Authorize.Net IDs** (Transaction ID, Batch ID, etc.) that are not in sequential order. If your system contains any functionality that accepts Authorize.Net generated IDs to be sequential, please update it immediately. Additionally, please make sure that your solution does not restrict any Authorize.Net ID field to 10 characters.