Understanding IIS Logging to Provide Leverage in Troubleshooting and Performance of Symantec Management Platform

Troubleshooting Altiris Performance Issues In IIS

- There are many things that can impact performance and functionality of the Altiris platform, the biggest of these being the misconfiguration of IIS.
- With the following paper we shall provide some information on how to properly utilize various features of IIS so that IIS is not a mystery component of the Altiris Platform.
- We will be focusing on Logs and how to properly utilize them to identify network problems, Symantec Agent communication issues, and IIS Configuration issues.

Using Logs to Troubleshoot Altiris Web Sites

- Reading HTTP Logs is important to find issues surrounding website page requests and connection problems.
  - HTTP or IIS Logs as they are commonly referred to, contain information regarding website traffic and requests to the pages within the web application.
  - They are located in the C:\IIS\Logs folder beginning in Windows Server 2008 and are written in the W3C extended log format by default.
  - They are customizable with various fields to record information regarding the web request.
  - Most of the fields are self-explanatory which include the following prefixes
    - S-Server Actions
    - C-Client Actions
    - Cs-Client to Server Actions
    - Sc-Server to client actions
- Error codes are updated and published via Microsoft in the following Article: https://support.microsoft.com/en-us/kb/943891
- The most common errors that are seen within IIS related to Altiris are 401, 404, and 503.
  - 401 is authorization related
  - 401.1 is a login failure. Usually an invalid username or password.
  - 401.2 is failure to login due to a server configuration. This is most often due to security settings such as Kerberos or a missing authentication method like Windows Authentication. Anonymous access is always listed and tried first within IIS and unless this is specifically removed, it will always be executed. This can create multiple entries in the IIS logs, first Anonymous then the proper credential passed. This credential or login account is listed in the logs.
  - 503 is a Service Unavailable. This is generally caused by a Service being stopped, such as the Altiris Service or World Wide Web Service. However, this can also mean that one of the AppPools for the Web site has stopped due to an overload of failed requests, the WAS service has prevented the worker process from restarting or even that the EventQueue processing tables have become full due to a SQL issue.

Reading and Understanding HTTPERR Logs

- HTTPERR logs contain information specific to the Application Pools
  - So if the connection request is returning a 503, you could check here to make sure the AppPool is running or if it has had its worker process stopped by the WAS Service due to a configuration problem. You can also look into the EventQueue table within SQL at this point to help narrow down an issue with event processing. This happens less and less with the improvements Symantec has made since 7.1 SP2 MP1.1, but still can be a reference point for troubleshooting.
  - They contain information related to responses to clients, connection time-outs, orphaned requests, and dropped connections that are handled incorrectly.
    - Responses to clients The HTTP API sends an error response to a client, for example, a 400 error that is caused by a parse error in the last received request. After the HTTP API sends the error
response, it closes the connection.
-Connection time-outs The HTTP API times out a connection. If a request is pending when the connection times out, the request is used to provide more information about the connection in the error log.
-Orphaned requests A user-mode process stops unexpectedly while there are still queued requests that are routed to that process. The HTTP API logs the orphaned requests in the error log.
-Specific error types are designated by Reason Phrase strings that always appear as the last field of each error line.

- Leveraging these logs can help determine where the problem is with relation to agent communication in Altiris.
- The most common Reason Phrases found in the HTTPERR logs are going to be Bad Request, Connection_Abandoned_By_AppPool, Timer_ConnectionIdle, Timer_EntityBody and Timer_AppPool.
  -Bad Request is noted by a 400 EventID in the HTTPERR logs.
  -This can have multiple meanings, but most commonly is that there is problem with the sending client system. The data being sent is malformed and the connection request is unable to be processed. Microsoft has commonly stated that this is caused by the NIC itself on the system or due to Network related issues.
  -Connection_Abandoned_By_AppPool is a problem that occurs when the worker process itself abandons or orphans a connection request. This could be a red herring as this happens frequently when there isn’t a timeout set for specific requests. However, it could also be a sign that there are more issues occurring within the Altiris program. Often a client will immediately make another request to connect if it receives this error on the IIS. Check Symantec Agent logs to further troubleshoot the connection issue and what process was attempting to occur.
  -Timer_ConnectionIdle is a basic event that actually doesn’t mean an error is occurring. The connection has simply expired and remains in an idle state. The AppPool has a setting itself, ConnectionTimeout, that is set to a default 2 Minutes.
  -Timer_EntityBody is an event caused by the request body failing to arrive before the connection expired. By default this is set to 2 Minutes and the timer is reset with another data indication received on the request.
  -Timer_AppPool. This is also tied to the ConnectionTimeout default setting of 2 Minutes, however the connection expired because the request waited too long for the AppPool Queue to process it. Adjusting the AppPool Queue as described above can take care of these events if they appear in the logs.

- You may also find Connection_Dropped_List_Full when the list of dropped connections is full. This is a newer event that Microsoft added in Windows Server 2008 and later.
  -HTTP/1.1 POST /ews/exchange.asmx - 1 Connection_Dropped_List_Full MSExchangeServicesAppPool
  -You can increase the AppPool Queue Length to address this issue using the following command
    -%systemroot%\system32\inetsrv\APPCMD set apppool “MyAppPool” /add.queueLength 10000

**ApplicationPools Settings**

- Classic .NET AppPool has a couple settings that can be adjusted that impact both the items found in the IIS and HTTPERR logs as well as performance. We mentioned the AppPool Queue Length above and here is another example of a setting that can be adjusted to improve performance.
  -Timeout
    -Adjust to 900 for increased recycle time. Adjusting this helps alleviate issues of connections staying open for periods of time
    -This was an older trick performed to help alleviate connection problems in IIS with Altiris SMP 7.1. This will depend on the environment though as this does mean that the recycle time for the Worker Process is shortened.
    -With the WAS Service introduction though, this does not need to be implemented as often as the WAS now controls the lifespan of the Worker Process.
  -In IIS 8.5, Windows Server 2012, there is a new feature that can be utilized to prevent this from happening, but can have a small impact upon performance
    -Idle Worker Process Page-out
      -When a worker process is not used for 20 minutes, IIS will terminate the process.
This is done to conserve memory and is very important on server where memory utilization is a concern. In IIS 8.5 instead of terminating the worker process when the Idle Time-out threshold is breached, you can Suspend it instead.

-When the worker process is requested for the first time the binaries need to be compiled, and for some web applications this can take some time. This compilation is often referred to as warming-up or starting-up the application. If you choose to Suspend the worker process when it times-out instead of terminating it, you can avoid this warm-up process from happening in this context.

*Note: This has not been validated by Symantec or ITS at this time, however, we would like to bring this to the attention of administrators.

WAS

- Windows Process Activation Service (WAS)
  - This is a new Service that was introduced in Windows Server 2008 R2 and can have an impact upon Application Pools.
  - The Windows Process Activation Service (WAS) manages the activation and lifetime of the worker processes that contain applications that host Windows Communication Foundation (WCF) services. The WAS process model generalizes the IIS 6.0 process model for the HTTP server by removing the dependency on HTTP. This allows WCF services to use both HTTP and non-HTTP protocols, such as Net.TCP, in a hosting environment that supports message-based activation and offers the ability to host a large number of applications on a given machine.
  - The impact now is that if an error occurs it can prevent the AppPool from restarting based on what has occurred. This will obviously prevent the web application from working properly and you will start seeing 503 Service Unavailable errors in the environment.
  - By default this Service is set to Manual, however it is still running and monitoring the worker processes of the Application Pools. Setting this to Automatic can reduce the likelihood of this preventing an automatic restart to an AppPool.

References

- https://support.microsoft.com/en-us/kb/820729
- https://support.microsoft.com/en-us/kb/943891

About the Author

David Winmill is a consulting engineer in Productive Systems Management at ITS Partners.