VIAVI Solutions

White Paper

Web Application Troubleshooting

Increasingly many business applications are being built with web frontends. In order to manage these applications, network managers require both solutions capable of isolating HTTP transaction details and providing analysis, as well as the knowledge to interpret the analysis and successfully resolve issues.

This paper provides foundational knowledge for IT professionals managing web applications by:

- Exploring the meaning of critical HTTP status codes
- Illustrating logical troubleshooting workflows
- Demonstrating how to provide evidence web developers can act upon

To ensure business apps run smoothly, network teams need solutions providing detailed HTTP analysis and the knowledge to quickly understand errors and resolve the issue.

HTTP Status Codes

In looking at web transactions and packets, it's first important to be familiar with the various response status codes. The following table outlines some of the more common HTTP status codes and what they mean to the network engineer.

Status Code	Description	Code Meaning
100	Continue	This could be indicative of a couple of things. Too many redirections could show some inefficiency internal to the web app design. It may also show some form of load balancing was being used.
200	OK	Standard response for successful HTTP request. This is what engineers want to see in web communications.
3xx	Redirection	This could be indicative of a couple of things. Too many redirections could show some inefficiency internal to the web app design. It may also show some form of load balancing was being used.
401	Unauthorized	Authentication either failed or was not provided.
403	Forbidden	Client issue of application making an illegal request. Server is refusing to respond.
404	Not Found	Requested resource was not found. Subsequent requests from client permissible.
500	Internal Server Error	Generic error for when server failed to fulfill a request.
502	Bad Gateway	Server was acting as a gateway or proxy, and received invalid response from upstream server.
503	Service Unavailable	Server currently unavailable; either overloaded or temporarily down. If server is under high memory pressure, it will temporarily refuse new requests. If this error code persists, server may need additional memory or memory configuration settings may need to be changed.

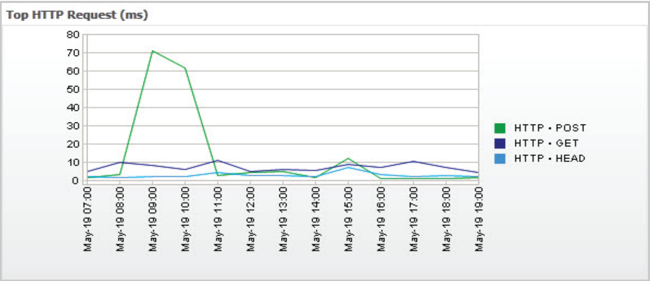
HTTP status codes show what is occurring within the app, and are valuable for app developers to resolve issues.

Establish Troubleshooting Workflows

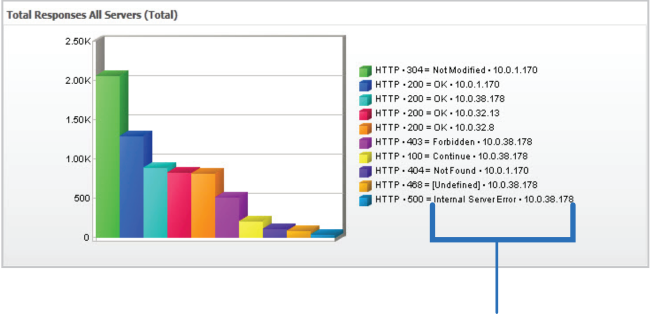
The HTTP status codes not only provide an indication to the network engineer of what might be happening within the application, they can also be valuable to the web application developer for resolving potential issues. But how do you isolate potential error codes?

In this example, a user complains that a web application is running slow, but it's not enough to understand that the server took too long to fulfill a user request. The engineer needs to drill down to isolate the exact server error and cause of the problem.

They begin by looking at a snapshot of HTTP Requests and Responses, which shows the type of requests that occurred during the timeframe represented in the horizontal graph. This is a first step in understanding what type of request and what response times occurred for each request.



In the next graph, different HTTP status code responses occurring during this same time period are listed. Looking at the status codes, notice that some of these are errors. Most notable is the "HTTP 500 = Internal Server Error" for 10.0.38.178. This type of 500 range error indicates the server is having trouble processing the request.

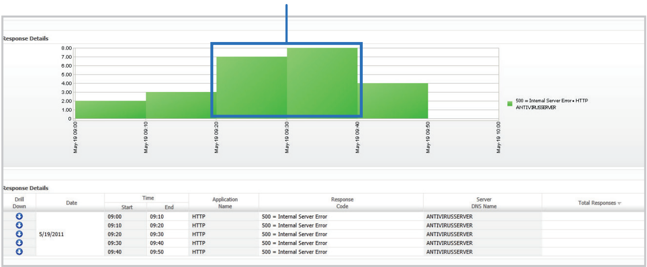


Error responsible for server slowdown

Long-term packet capture and logical workflows are key to quickly navigating to application errors.

Investigating further, this is where it’s important to have both trending and long-term packet capture technologies in place to see how prevalent the problem is and isolate the exact cause. With trending activated, the engineer can see when and how often server errors occurred. The trouble ticket indicates the user complaint was between 9:00 am and 9:40 am. The report showed the frequency of errors, and identified the problem server for this time period.

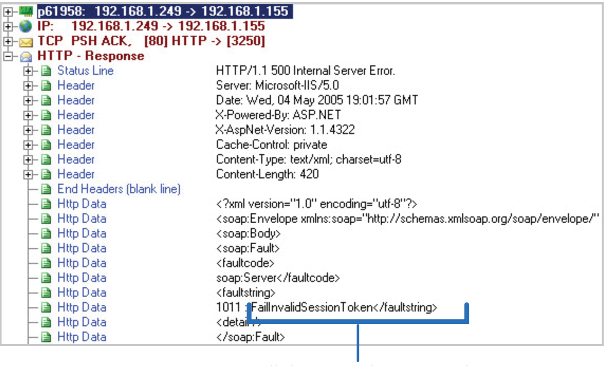
Periods with greatest number of errors



Response Details

Drill Down	Date	Time		Application Name	Response Code	Server DNS Name	Total Responses
		Start	End				
⬇	5/19/2011	09:00	09:10	HTTP	500 = Internal Server Error	ANTIVIRUSSERVER	
⬇		09:10	09:20	HTTP	500 = Internal Server Error	ANTIVIRUSSERVER	
⬇		09:20	09:30	HTTP	500 = Internal Server Error	ANTIVIRUSSERVER	
⬇		09:30	09:40	HTTP	500 = Internal Server Error	ANTIVIRUSSERVER	
⬇		09:40	09:50	HTTP	500 = Internal Server Error	ANTIVIRUSSERVER	

From within this report, you can drill down and mine the packets associated with this server and condition and analyze them for the specific issue. In this case, the process is automated using the Observer® Platform. With the reporting capabilities available as a part of the Apex™ dashboards, simply drill down to GigaStor™ and mine the packets associated with the problem.



```
p61958:  192.168.1.249 -> 192.168.1.155
IP:    192.168.1.249 -> 192.168.1.155
TCP  PSH ACK,   [80] HTTP -> [3250]
HTTP - Response
   Status Line               HTTP/1.1 500 Internal Server Error.
   Header                    Server: Microsoft-IIS/5.0
   Header                    Date: Wed, 04 May 2005 19:01:57 GMT
   Header                    X-Powered-By: ASP.NET
   Header                    X-AspNet-Version: 1.1.4322
   Header                    Cache-Control: private
   Header                    Content-Type: text/xml; charset=utf-8
   Header                    Content-Length: 420
   End Headers (blank line)
   Http Data                 <?xml version="1.0" encoding="utf-8"?>
   Http Data                 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
   Http Data                 <soap:Body>
   Http Data                 <soap:Fault>
   Http Data                 <faultcode>
   Http Data                 soap:Server</faultcode>
   Http Data                 <faultstring>
   Http Data                 1011 :FailInvalidSessionToken</faultstring>
   Http Data                 <detail />
   Http Data                 </soap:Fault>
```

Drill down to the cause of the error

In this situation, the error was caused by a Failed Session Token. There is a host of other information that will help the web application designers resolve the issue and eliminate this particular problem from reoccurring.

With a comprehensive understanding of HTTP status codes and a logical troubleshooting workflow, network managers are able to resolve web application errors quickly and provide actionable information to web developers to effectively solve the problem.

VIAVI Solutions

Contact Us **+1 844 GO VIAVI**
(+1 844 468 4284)

To reach the VIAVI office nearest you,
visit viavisolutions.com/contact

webapplicationprimer-wp-ec-ae
30176224 901 0914