White Paper

Testing DNS Servers Using the ClearSight Packet Generator

This write-up describes how to use the ClearSight Packet Generator and ClearSight Analyzer in combination to test Domain Name Servers. The packet generator is used to transmit a trace file that contains previously captured DNS queries to a specific DNS server. The ClearSight Analyzer is then used to monitor the traffic looking for slow DNS responses and failed responses.

Table of contents

FLUKE networks

Testing DNS Servers:

A reliable DNS server is critical to providing responsive applications. If the DNS server takes a long time to respond to DNS queries or does not respond at all, the overall application will appear slow. Problems receiving responses to the queries can be the result of heavily loaded servers, a misconfigured DNS server, or packet loss between the device sending the query and the server.

One method of testing DNS servers is to use commands such as NSLookup to perform a DNS query. While this method works, it is very time consuming when testing a server using multiple domain names and does not lend itself to running the test multiple times.

The ClearSight Packet Generator provides a means of actively sending a set of DNS queries to any DNS server. The ClearSight Analyzer can then be used to monitor the response to each one of these queries. There are three methods available to generate the DNS queries that will be sent to the DNS server. The first method is to create a packet from scratch. The second method is to import a trace file containing previously captured DNS frames and change the source and destination addresses. The third method is to use the scripting function to transmit a previously captured trace file and change the addressing information automatically.

For this test we selected the third method, using a script to change the addressing information and transmit the trace file. The scripting function allowed us to control the following variables:

- The trace file containing the DNS queries
- The Source IP Address
- The Source Port Number
- The Destination IP Address (the IP address of the DNS server to be tested)
- The number of times the test was to be repeated
- The MAC address of the default router

In this case we chose to run the test 30 times. Each time we used our PC's IP address and selected a random source port for each query. The test was run against one of AT&T's DNS servers.

The following script was created and placed in the jsfiles directory located in the ClearSight Packet Generator program directory. The file was given an extension of **.js**.

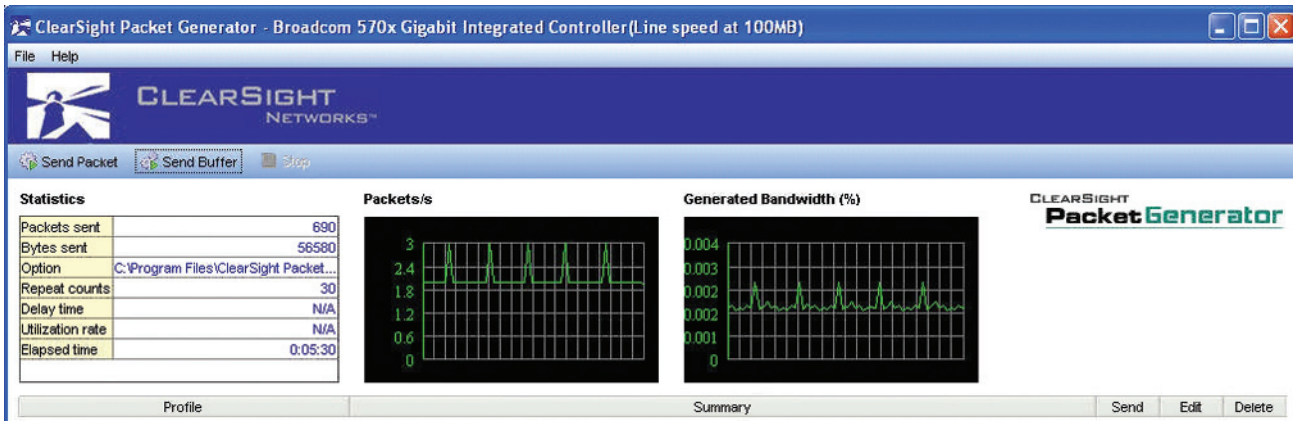
Example script:

```
var work_dir = java.lang.System.getProperty("user.dir");

var tracefile = work_dir + "\\traces\\dnstraffic1.enc";

var apply_gap = true;

var buffer_times = 30;

var src_ipaddr = "10.0.0.202";

var rand_src_port = true;

var dst_ipaddr = "12.127.16.68";

var appDancer = getAppDancerObjectModel();
```

```
send_edit_buffer();

function send_edit_buffer()

{

var pktGen = appDancer.getPktGenerator();

if (pktGen.isSending())

pktGen.stop();

var options = pktGen.getSendBufferOptions();

options.setTracefileName(tracefile);

options.setGap(apply_gap);

options.setBufferTimes(buffer_times);

options.setRouterMacAddr(18706471717);

pktGen.processConnectionsInBuffer();

var conns = options.getConnections();

for (i = 0; i < conns.length; i++)

{

var conn = conns[i];

var srcIPAddr =

Packages.com.appdancernet.appdancer.shared.netutils.IpUtil.IpAddrStrToRaw(src_ipaddr);

conn.setSrcIPAddr(srcIPAddr);

conn.setSrcPortRandom(rand_src_port);

var dstIPAddr =

Packages.com.appdancernet.appdancer.shared.netutils.IpUtil.IpAddrStrToRaw(dst_ipaddr);

conn.setDstIPAddr(dstIPAddr);

}

pktGen.sendBuffer();

}
```

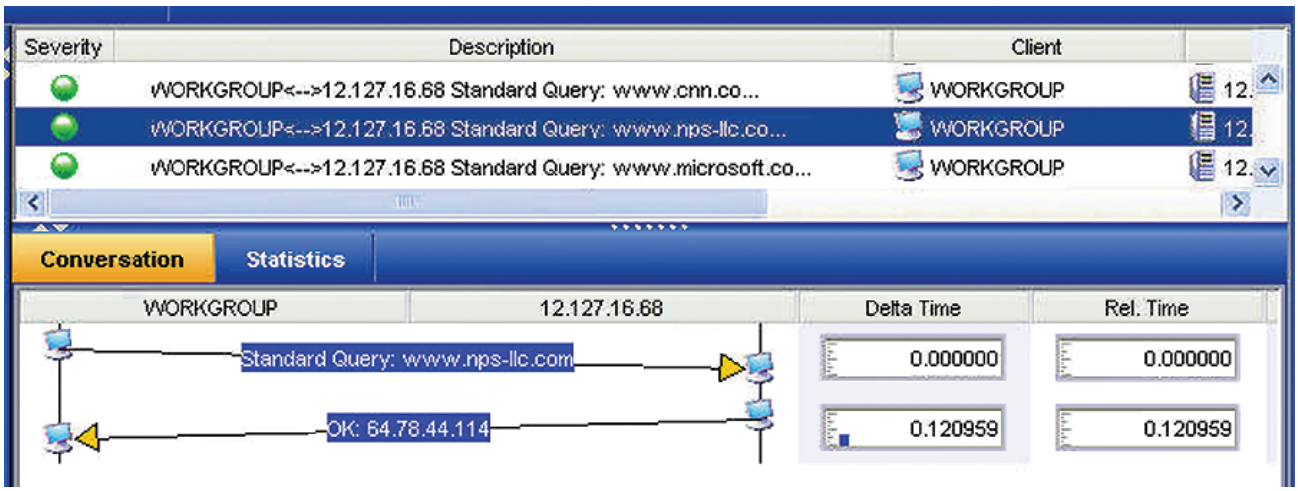
After the script was created, it was run by selecting **File – Run Script** from the packet generator's menu. The packet generator then began sending each one of the queries using the addressing information specified in the script file. While the script was run, the ClearSight Analyzer was used to monitor the network traffic.

While the script is running the packet generator displays the rate at which the transmitted frames are being presented to the network. In this case each query contained in the trace file has an interframe gap of 500 milliseconds. The resulting frame rate is 2 frames per second. The **apply_gap** variable in the script determines whether the interframe gap contained in the trace file will be used, or whether the frames will be sent as fast as the packet generator can transmit them. If this value is set to false, the packet generator will attempt to transmit each query at the line rate of the network interface card selected. Setting the **apply_gap** variable to **false** is a method of sending queries to the server at a very high rate to stress test the server.



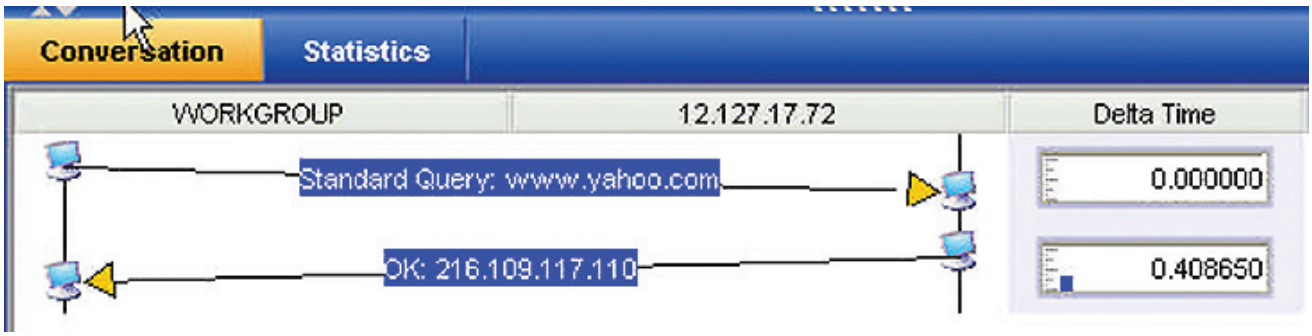
While the packets were being transmitted by the packet generator, the network traffic was monitored using the ClearSight Analyzer. The analyzer tracks each query, monitoring the result of the query and the query response time. This information can be used to determine if certain queries take longer than others or if they result in a DNS failure.

The following screen shows the DNS queries sent by the ClearSight Packet Generator. For each query we can graphically see the DNS request and the response from the DNS server.

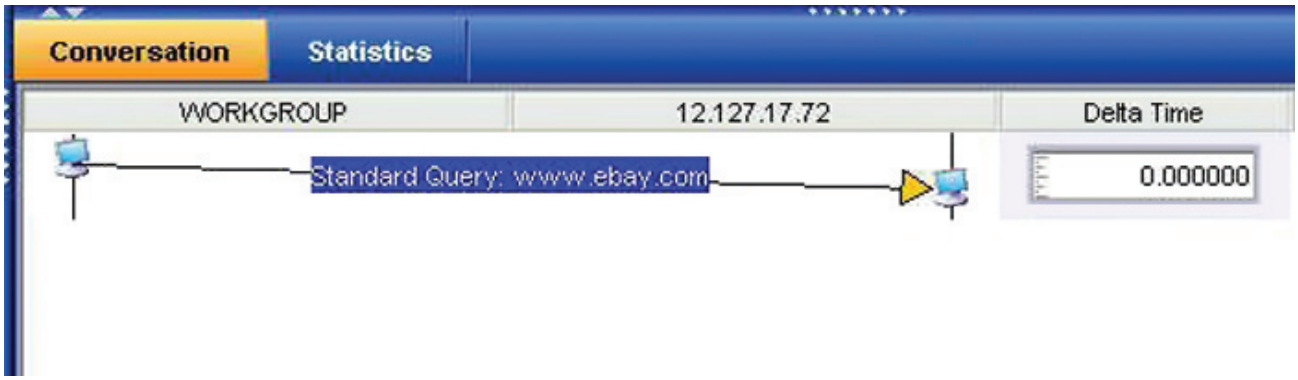


In this case, a DNS query is sent for the A record www.nps-llc.com. The DNS server responds to this query in 120.959 milliseconds. It is important to note that if an application were to be accessing this site, the application cannot proceed until it receives a response to the query. If the server takes a long time to respond, the application will appear to be slow to the person using the application.

The script was modified to run the queries using a different DNS server. In this case, we began to see both long response times and cases where there was not response at all. Both of these situations can result in an overall increase in application response time.



In the example above a query was sent to the DNS server for www.yahoo.com. The server took 408 milliseconds to respond to this query. While we did get a response, it took longer than normal. This could be a result of the load on the server or congestion on the network between our PC and the DNS server.



The query above was sent, but a response was never received. Most DNS resolvers will wait 2 seconds before resending the request. This would have resulted in at least a 2 second delay before the application would be able to contact the web server.

The ClearSight Analyzer could have been used to passively monitor DNS queries on the network without the use of the packet generator. However, the results would have been based on those queries sent during the time we were monitoring. By actively creating our own DNS queries, we were able to send enough traffic to get a good picture of the server's ability to receive, process and reply to our requests.

This example shows how the ClearSight Packet Generator can be used to test a DNS server. It can also be used to generate a variety of other types of traffic as well. NetBIOS queries can be captured and retransmitted to test the capabilities of WINS servers. TCP SYN frames used to ensure firewalls are passing allowed ports and discarding those ports that have been blocked. The use of the TCP SYN frames is one method of performing regression testing on the firewall after a change has been made to the firewall rules.

About Network Protocol Specialists, LLC:

Mike Pennacchi (mike@nps-llc.com) is owner of Network Protocol Specialists, a network analysis and training company based in Seattle Washington. His company specializes in analyzing network performance problems for companies throughout the United States. Mike has been a speaker at Networld+InterOp since 1997 and has received the Highest Satisfaction with the Instructor award two of those years. Mike brings his experience as a network analyst into the classroom and assists the students in understanding how to fix problems in their own networks.

networkprotocol
specialists