Protocol analyzers can make short work of admin tasks

Apr 18, 2002
Ron Nutter

It used to be that protocol analyzers were expensive pieces of hardware, costing upwards of $20,000 and requiring specialized training to use. However, things have changed quite a bit in just a few years. Protocol analyzers now top out at around $1,000, and some are even free—and they can all make the life of a network administrator much easier. I'm going to explain how you can use various protocol analyzers on your network to perform such tasks as benchmarking, intrusion detection, and troubleshooting e-mail problems.

Finding network abnormalities

I have never used a protocol analyzer for a byte-level analysis to resolve a problem. Instead, I usually use one to benchmark my network or to spot abnormalities when troubleshooting. For example, several years ago, I received a panicked phone call from a network administrator in a bank several hours away from the office where I worked. Its network was locking up every 10 to 15 minutes. I talked with the administrator for several minutes and had him make sure that other possible causes such as a bad electrical ground, faulty network cable, or a broken network card weren't the source of the problem.

After I arrived at the site, I ran the protocol analyzer for a few minutes. It was then that I noticed something strange: Each workstation on the network was requesting the current date and time from the Novell server 20 to 30 times per minute. In normal conditions, this should happen only when the workstations boot up. A little investigation found that a third-party utility was being loaded that was supposed to get the current date and time about two or three times per day. After removing this utility from the workstations, the problem disappeared. Had I not been using a protocol analyzer, my troubleshooting time would have been much longer.

Perform intrusion detection

Unfortunately, detecting intrusions is becoming more and more important as unwelcome visitors from the outside try to access and damage your network. This is another area where a protocol analyzer can be handy. First, look for services that shouldn’t be running on a particular server, such as FTP. It's a good practice to check for and disable such rogue services whenever new servers are added to your network and when service packs or updates are applied to existing servers.

You should also watch for people trying to do things that they shouldn't be doing on your servers. For example, say you have a server that allows *you* to use the Secure Shell utility for remote administration. Upon analyzing the server, you find another user taking advantage of this open port (ssh or port 22). This allows you to immediately track down their source address and block that address from accessing your network. Another way to find intrusions is to look at login accounts that have been disabled or should have been disabled to see whether they are being used to access the network.

Check for virus activity

Several protocol analyzers (EtherPeek and Sniffer, for example) offer the ability to download filters that let you

view specific types of traffic on your network. Instead of having to sort through all the network traffic, you can just download predefined filters to scan for virus activity such as Code Red and Nimda. I like to run these filters in what I call a global mode, which looks at all the packets crossing the wire regardless of source or destination.

You can also create your own virus filters. The information you need is contained in the virus alerts issued by such companies as McAfee and Norton. Looking for a file attachment by name in a mail message or looking for a certain command on an HTTP header line are just a couple of ways you can take a more proactive stance toward virus protection.

Watch out for unauthorized programs

With the IP-based network and the Internet becoming commonplace, it's easier to find unauthorized programs on your network and stop their use. The proliferation of peer-to-peer file sharing applications such as BearShare and Napster has consumed network bandwidth that could be better used elsewhere. The best way to halt usage of such applications is to download the applications onto a test workstation and have a protocol analyzer watch for traffic going to and coming from the IP address of the test workstation. Once you've seen the traffic created, you can create filters that stop the application's usage. Each analyzer has a different method for creating such filters, so you will want to take a look at your application's documentation for this step.

Check for WAN link usage

When you have more than one T1 connection to the Internet, knowing these links are working correctly is critical to the health of your network. If routing protocols such as OSPF and BGP4 are being used, it can be helpful to be able to see what the problem is when things go awry. Not all protocol analyzers can track all IP traffic patterns, so knowing what is required to monitor your T1 or similar link can help decide what analyzer will be best for you.

One tool that can track patterns is the Sniffer Portable WAN tool. This high-end utility automatically finds and labels Internetwork problems such as retransmissions, duplicate IP addresses, high rate of physical cyclic redundancy check (CRC) errors, WAN overload, and frame relay congestion. Once an issue is detected, Sniffer recommends solutions to potential network problems.

Many enterprise-level analyzers require special PCMCIA cards with the appropriate type of connectors to sit in series with the V.35 or other type of connector that your laptop or workstation may use. For nonportable solutions, you may end up getting either an external pod-like interface or a special interface board to go into a conventional desktop form factor. This same process also applies to ATM and DS3 connections.

Check for e-mail problems

I use protocol analyzers to monitor e-mail problems much more than I would have thought. To do this, you must set up an analyzer with a filter that monitors the IP ports used by a mail server (typically port 25 for SMTP, 110 for POP3, and 143 for IMAP) to send and receive mail. Several good examples of how to do this are on packet-level.com.

I've found the type of filter I described above to be useful in figuring out why a particular e-mail won't go through when the only error I get in the Exchange server logs is "communications error." I have made the modification to the filter that the site suggests, but this modification just examines e-mail to and from a particular mail server. However, this technique is still a big help because I don't have to go through an entire capture session to look for the mail traffic. Entire capture sessions can be quite large, depending on the size of your network.

Verify that your firewall is working correctly

Since firewalls protect your network from unwelcome visitors, knowing that they're working correctly is important for verifying the security of your network. Checking the firewall will involve using several different filters (these can be predefined filters, administer-created filters, or downloaded filters, all with various functions), depending on the level of sophistication of the packet filtering being used.

In general, you will have two sets of filters, one checking packets based on outgoing traffic and one based on incoming traffic. Leaving the incoming filter running 24/7 would be a good idea, because this filter will be a good indication that the firewall is working as expected and will provide a quick alert if the firewall fails for some reason and begins letting unwanted packets through.

For example, NetDoppler utilizes several features of the ICMP, IP, and DNS protocols to perform tasks and tests on remote hosts to check latency and throughput and to isolate problems. PacketScrubber removes sensitive or confidential data from frames and packets within a trace file by changing the packet and frame payloads to null data.

Summary

We've just scratched the surface of the possible uses of a protocol analyzer. Before you go out and buy the first one you see or purchase something that a vendor recommends, try to obtain trial versions of a few, use them, and see which candidate best meets your needs. It's also a good idea to keep the analyzer you buy under some type of maintenance contract from the vendor to keep the application up to date and problem-free.