Realtime publishers

The Definitive Guide™ To

Active Directory Troubleshooting, Auditing, and Best Practices

2011 Edition

Don Jones

Copyright Statement

Chapter 2: Monitoring Active Directory

The fact is that you can't really do anything with Active Directory (AD) unless you have some way of figuring out what's going on under the hood. That's what this chapter will be all about: how to monitor AD. I have to make a distinction between *monitoring* and *auditing:* Monitoring, which we'll cover here, is primarily done to keep an eye on functionality and performance, and to solve functional and performance problems when they arise. Auditing is an activity designed to keep an eye on what people are *doing with* the directory—exercising permissions, changing the configuration, and so forth. We have chapters on auditing lined up for later in this book.

Monitoring Goals

There are really two reasons to monitor AD. The first is because there's some kind of problem that you're trying to solve. In those cases, you're usually interested in current information, delivered in real-time, and you're not necessarily interested in storing that data for more than a few moments. That is, you want to see what's happening *right now.* You also usually want to focus in on specific data, such as that related to replication, user logon performance, or whatever you're troubleshooting.

The second reason to monitor is for trending purposes. That is, you're not looking at a specific problem but instead collecting data so that you can spot potential problems. You're usually looking at a much broader array of data because you don't have anything specific that you need to focus on. You're also usually interested in retaining that data for a potentially long time so that you can detect trends. For example, if user logon workload is slowly growing over time, storing monitoring data and examining trends—perhaps in the form of charts—allows you to spot that growing trend, anticipate what you might need to do about it, and get it done.

Having these goals in mind as we look at some of the available tools is important. Some tools excel at offering real-time data but are poor at storing data that would provide trending information. Other tools might be great at storing information for long-term trending but aren't as good at providing highly-detailed, very-specific, real-time information for troubleshooting purposes. So as we look at these tools, we'll try to identify which bits they're good at.

Another thing to keep in mind before we jump in is that some of these tools are actually foundational technologies. In other words, when we discuss event logs, you have to keep in mind that that technology is a tool that you can use—and it's a foundation that *other* tools use. Any strengths or weaknesses present in that technology are going to carry through to any tools that *use* that technology. So again, it's simply important to recognize such considerations because they'll have an impact beyond that specific tool.

Event Logs

Windows' native event logs play a crucial role in monitoring AD. The event logs aren't great, but they're the place where AD sends a decent amount of diagnostic and auditing information, so you have to get used to using them.

There's a bit of a distinction that needs to be made: The *event log* is a native Windows data store. The *Event Viewer* is the native tool that enables you to look at these logs. Event logs themselves are also accessible to a wide variety of other tools, including Windows PowerShell, Windows Management Instrumentation (WMI), and numerous third-party tools. In Windows Server 2008 and later, these logs' Viewer is accessible through the Server Manager console, which Figure 2.1 shows.

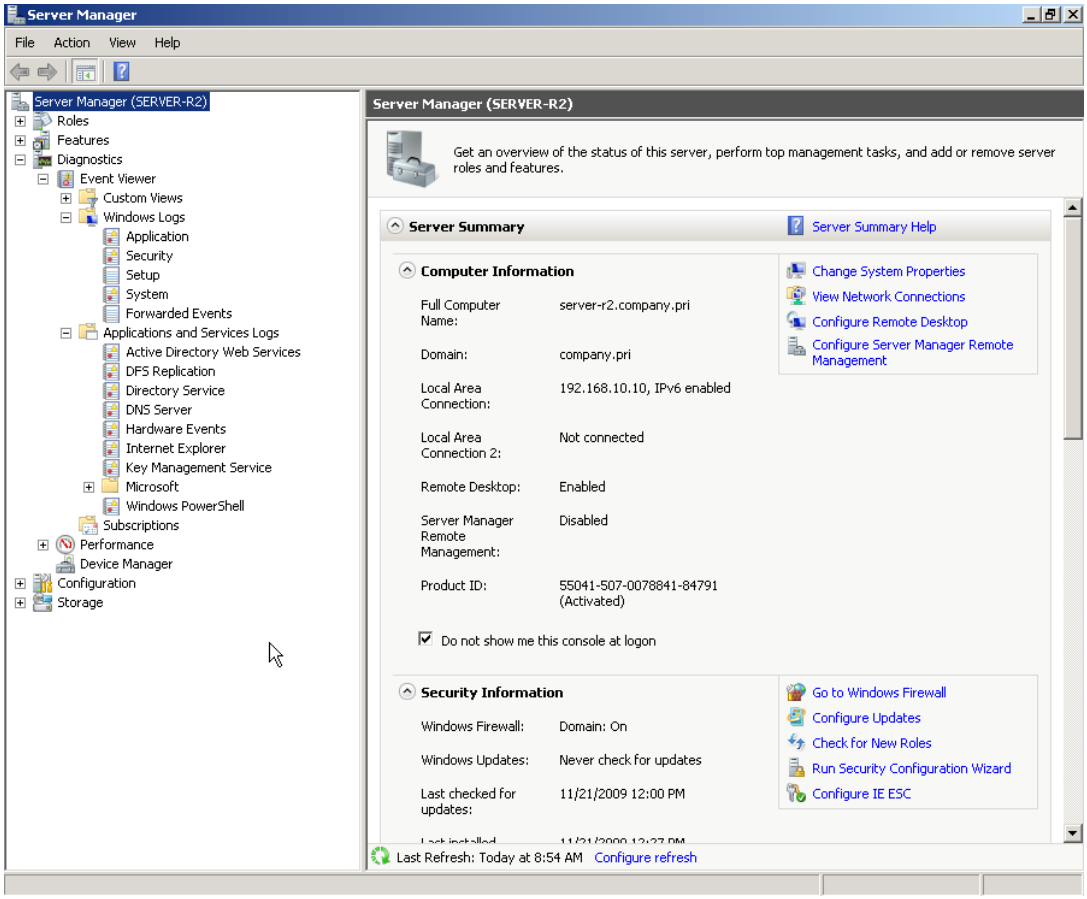

Figure 2.1: Accessing event logs in Server Manager.

There are two kinds of logs. The *Windows Logs* are the same basic logs that have been around since the first version of Windows NT. Of these, Active Directory (AD) writes primarily to the Security log (auditing information) and the System log (diagnostic information). In Windows Server 2008, a new kind of log, *Applications and Services Logs,* were introduced. These supplement the Windows Logs by giving each application the ability to create and write to its own log rather than dumping everything into the Application log, as was done in the past. In these new logs, AD creates an Active Directory Web Services log, DFS Replication log, Directory Service log, and DNS Server log. Technically, DFS and DNS aren't part of AD, but they do integrate with and support AD, so they're important to look at.

Windows itself also creates numerous logs under the Microsoft folder, as Figure 2.1 shows: GroupPolicy, DNS Client Events, and a few others, all of which can offer clues into AD's operation and performance. Don't forget that client computers play a role in AD, as well. Logs for NTLM, Winlogon, DNS Client, and so forth can all provide useful information when you're troubleshooting an AD problem.

Although the event logs can contain a wealth of information, their usefulness can be hit or miss. For example, the event that Figure 2.2 shows is pretty clear: Smart card logons aren't working because there isn't a certificate installed. My domain doesn't use smart card logons, so this is expected and doesn't present a problem.

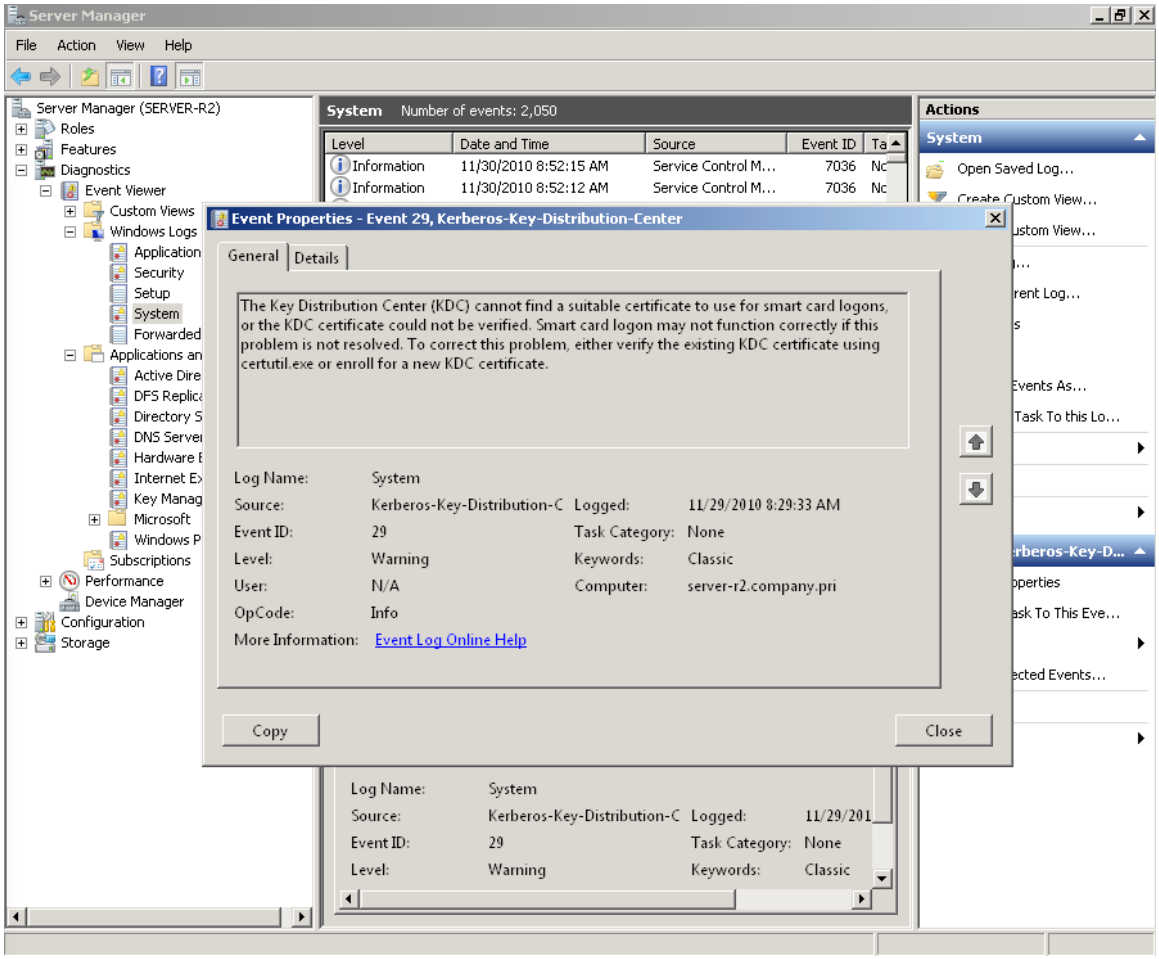

Figure 2.2: Helpful events.

Other events just constitute "noise," such as the one shown in Figure 2.3: User Logon Notification for Customer Experience Improvement Program. Huh? Why do I care?

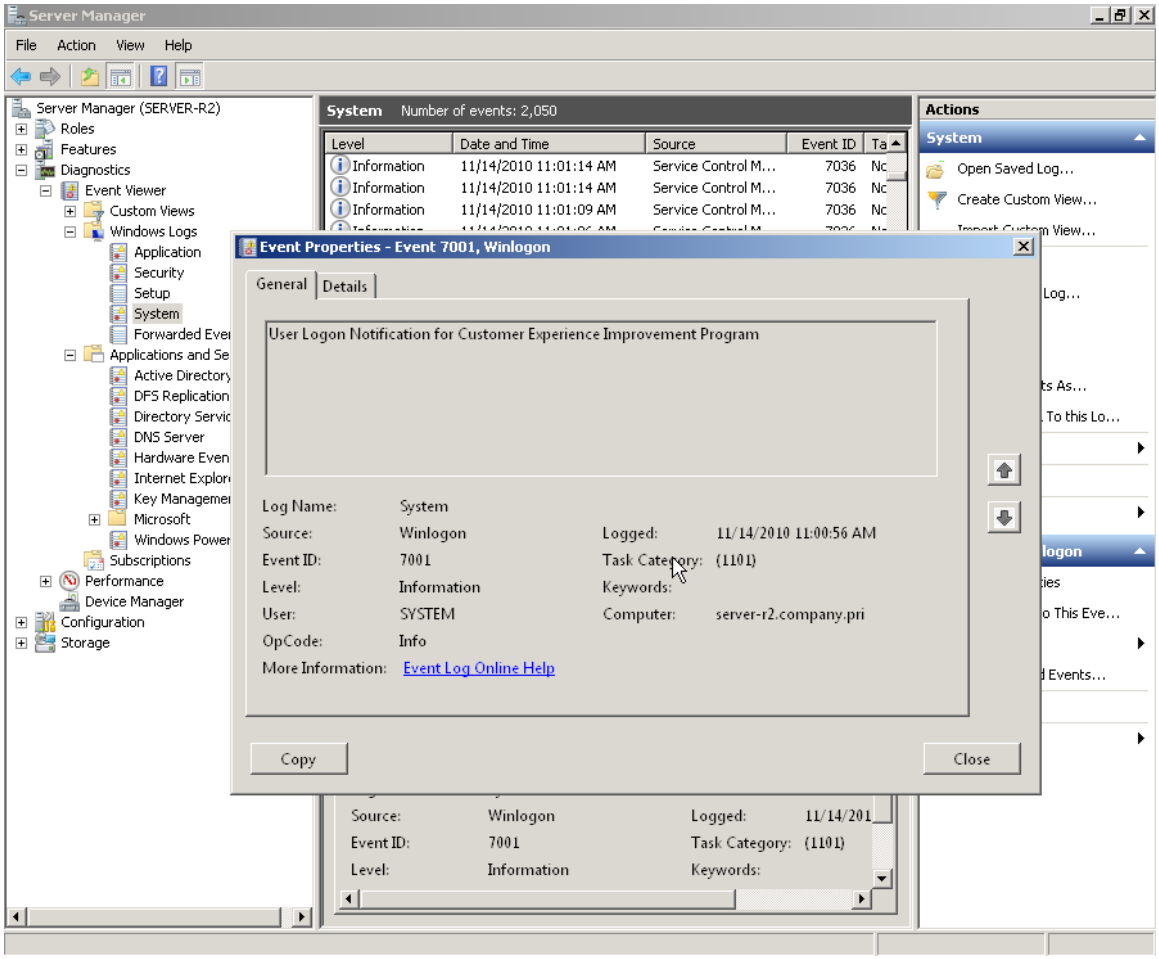

Figure 2.3: "Noise" events.

Then you've got winners like the one shown in Figure 2.4. This is tagged as an actual error, but it doesn't tell me much—and it doesn't give many clues about how to solve the problem or even if I *need* to worry about it.

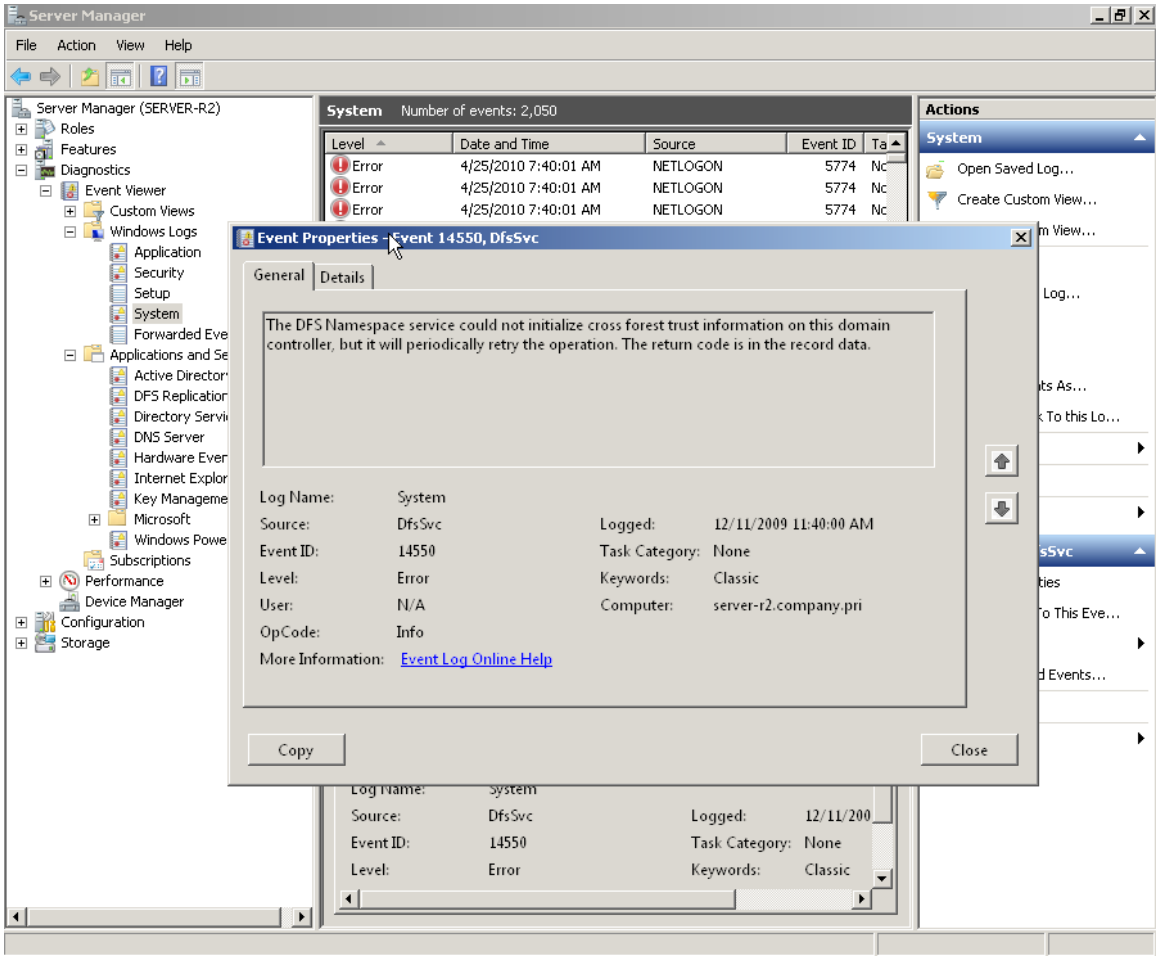

Figure 2.4: Unhelpful events.

It's probably going too far to call this event "useless," but this event is certainly not very helpful. Finally, as shown in Figure 2.5, sometimes the event logs will include suggestions. That's nice, but is this the best place to put these? They create more "noise" when you're trying to track down information related to a specific problem, and they're tagged as Warnings (so you tend to want to look at them, just in case they're warning you of a problem), but they can often be ignored.

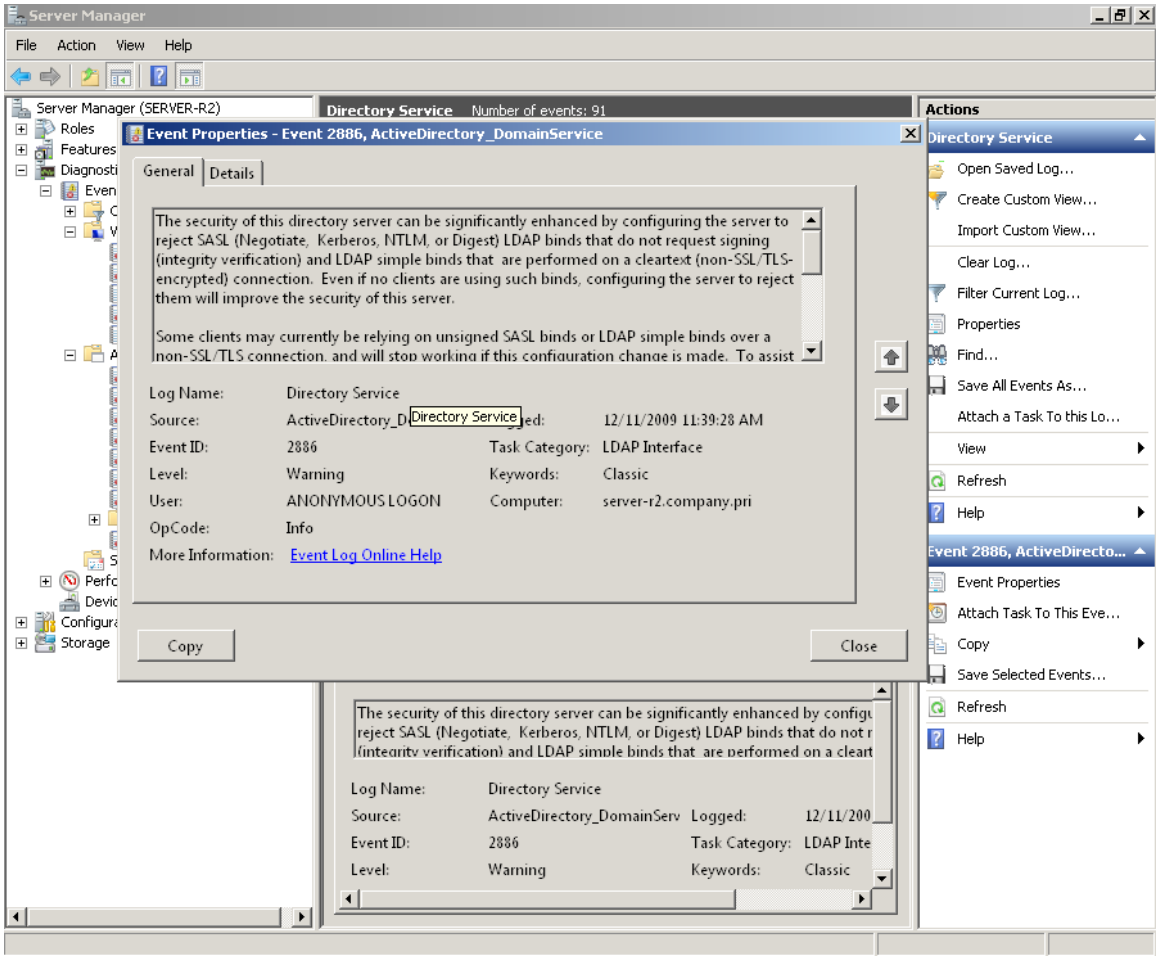

Figure 2.5: Suggestions, not "events."

There probably isn't an administrator alive who hasn't spent a significant amount of time in Google hunting down the meaning behind—and resolution for—dozens of event IDs over the course of their careers. That reality highlights key problems of the native event logs:

- They're not centralized. Although you can configure event forwarding, it's pretty painful to get all of your domain controllers' logs into a single location. That means your diagnostic information is spread across multiple servers, giving you multiple places to search when you're trying to solve a problem.
- They're not always very clear. Confusing, vague, or obtuse messages are what the event logs are famous for. Although Microsoft has gradually improved that over the years in some instances, there are still plenty of poor examples in the logs.
- They're full of noise. Worse, you can't rely on the "Information," "Warning," and "Error" tags. Sometimes, an "Information" event will give you the clue you need to solve a problem, and "Warning" events—as we've seen—can contain information that is not trouble-related.
- The native Viewer tool offers poor filtering and searching capabilities, and no correlation capability. That is, it can't help you spot related events that might point to a specific problem or solution.

Problems notwithstanding, you *have* to get used to these logs because they're the *only* place where AD and its various companions log *any* kind of diagnostic information when problems occur.

System Monitor/Performance Monitor

Also located in Server Manager is Performance Monitor, the native GUI-based tool used to view Windows' built-in performance counters. Any domain controller will contain numerous counter sets related to directory services, including several DFS-related categories, DirectoryServices, DNS, and more. These are designed to provide the focused, real-time information you need when you're troubleshooting specific problems—typically, performance problems, although not necessarily. Although Performance Monitor does have the ability to create logs, containing performance data collected over a long period of time, it's not a great tool for doing so. More on that in a bit.

It's difficult to give you a fixed list of counters that you should always look at; *any* of them might be useful when you're troubleshooting a specific problem. That said, there are a few that are useful for monitoring AD performance in general:

- DRA Inbound Bytes Total/Sec shows inbound replication traffic. If it's zero, there's no replication, which is generally a problem unless you have only one domain controller.
- DRA Inbound Object Updates Remaining in Packet provides the number of directory objects that have been received but not yet applied. This number should always be low on average, although it may spike as replicated objects arrive. If it remains high, your server isn't processing updates quickly.
- DRA Outbound Bytes Total/Sec offers the data being sent from the server due to replication. Again, unless you've got only one domain controller, this will rarely be zero in a normal environment.
- DRA Pending Replication Synchronization shows the number of directory objects waiting to be synchronized. This may spike but should be low on average.
- DS Threads in Use provides the number of process threads currently servicing clients. Continuously high numbers suggest a need for a larger number of processor cores to run those threads in parallel.
- Kerberos Authentications offers a basic measure of authentication workload.
- LDAP Bind Time shows the number of milliseconds that the last LDAP bind took to complete. This should be low on average; if it remains high, the server isn't keeping up with demand.
- LDAP Client Sessions is another basic unit of workload measurement.
- LDAP Searches/Sec offers another good basic unit of workload measurement.

All of these counters benefit from trending, as they all help you form a basic picture of how busy a domain controller is. In other words, it's great when you can capture this kind of data on a continuous basis, then view charts to see how it changes over time. Performance Monitor itself isn't a great tool for doing that because it simply wasn't designed to collect weeks and weeks worth of data and display it in any meaningful way. However, it can be suitable for collecting data for shorter periods of time—say, a few hours—then using the collected data to get a sense of your general workload.

You'll have to do that monitoring on *each* domain controller, too, because the performance information is local to each computer. Ideally, each domain controller's workload will be roughly equal. If they're not, start looking at things like other tasks the computer is performing, or the computer's hardware, to see why one domain controller seems to be working harder than others.

This kind of performance monitoring is one of the biggest markets for third-party tools, which we'll discuss toward the end of this chapter. Using the same underlying performance counters, third-party tools (as well as additional, commercial tools from Microsoft) can provide better performance data collection, storage, trending, and reporting—and can even do a better job of sending alerts when performance data exceeds pre-set thresholds. What Performance Monitor is good at—as Figure 2.6 shows—is enabling you to quickly view real-time data when you're focusing on a specific problem.

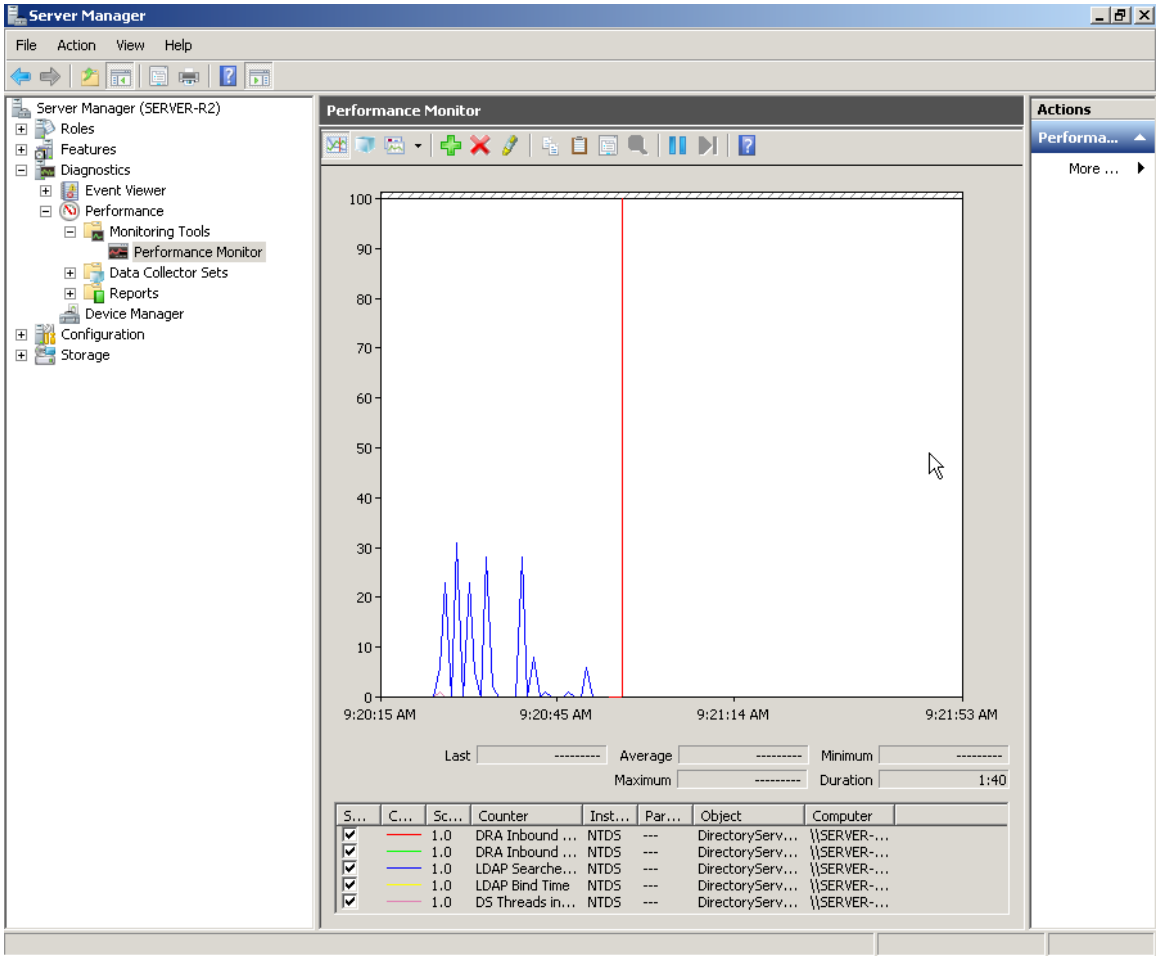

Figure 2.6: Viewing real-time performance data in Performance Monitor.

One problem we should identify, though, is that Performance Monitor requires a good deal of knowledge on your part to be useful. First, you have to make sure you're looking at all the right counters at the right time. Looking at DS Threads alone is useless unless you're also looking at some other counters to tell you *why* all those threads are, or are not, in use. In other words, you have to be able to mentally correlate the information from many counters to get an accurate assessment of how AD is really performing. Microsoft helps by providing predefined *data collector sets,* which can include not only counters but also trace logs and configuration changes. One is provided for AD diagnostics (see Figure 2.7).

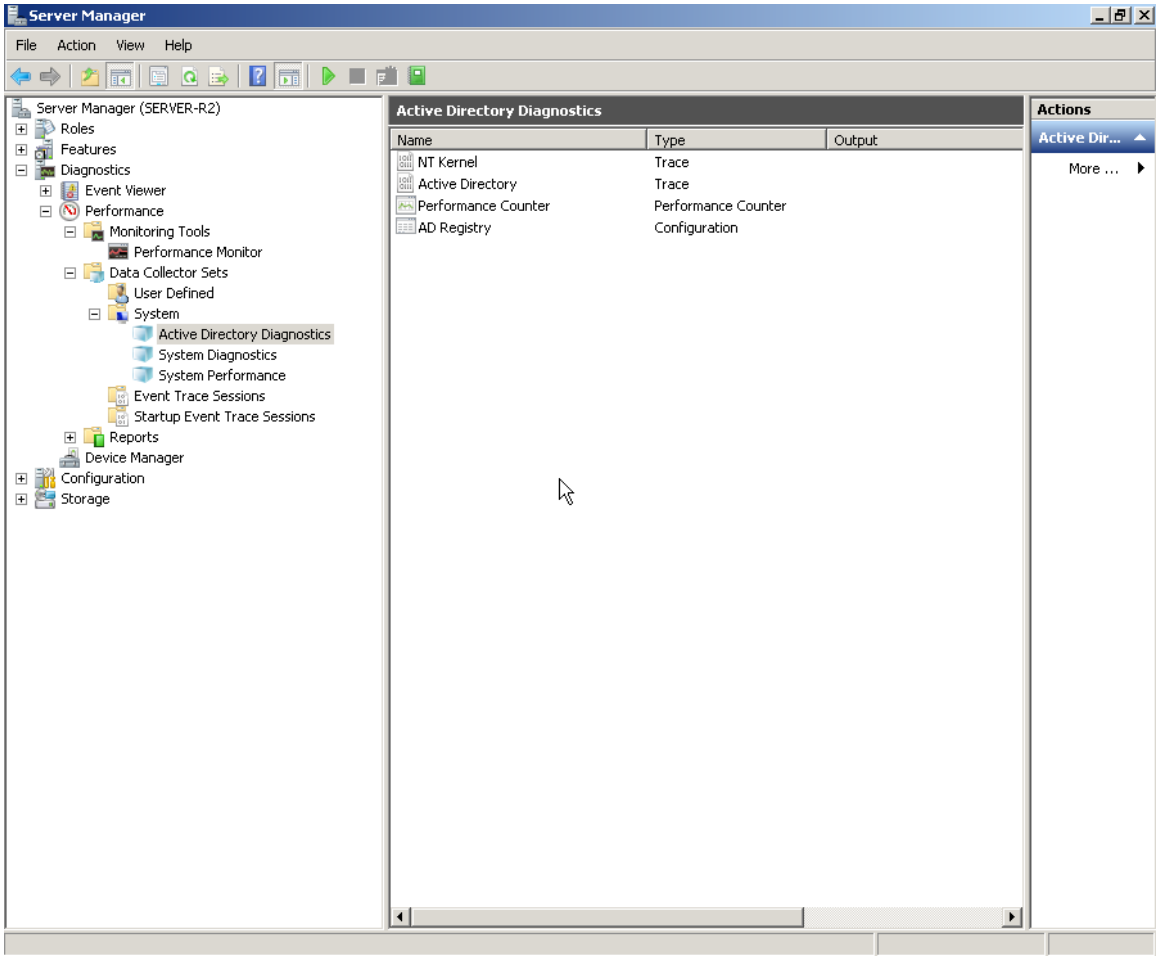

Figure 2.7: The AD Diagnostics data collector set.

Once you start a collector set, you can let it run for however long you like. Results aren't displayed in real-time; instead, you have to view the latest report, which is a snapshot. These sets are designed to run for longer periods of time than a normal counter trace log, and the sets' configuration includes settings for managing the collected log size. Figure 2.8 shows an example report.

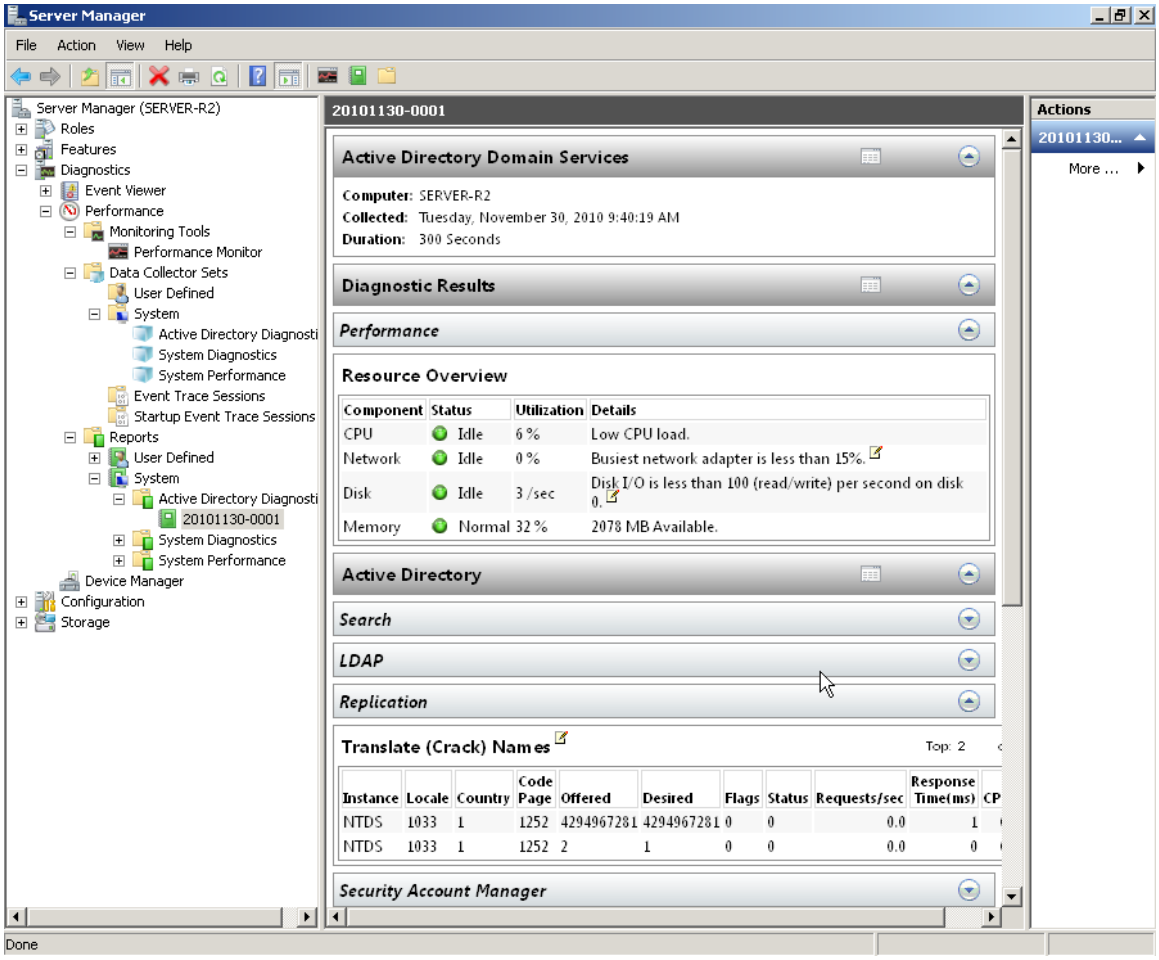

Figure 2.8: Viewing a data collector set report.

These reports do a decent job of applying some intelligence to the underlying data. As you can see here, a "green light" icon lets you know that particular components are performing within Microsoft's recommended thresholds. That "intelligence" doesn't extend far, though: Once you start digging into AD-specific stuff, you're still looking at raw data, as you can see in the section on Replication that's been expanded in Figure 2.8. Thus, you'll still need a decent amount of expertise to interpret these reports and determine whether they represent a problem condition.

Command-Line Tools

A host of command-line tools can help detect AD problems or provide information needed to solve those problems. This chapter isn't intended to provide a comprehensive list of them, but one of the more well-known and useful ones includes Repadmin. This tool can be used to check replication status and diagnose replication problems. For example, as Figure 2.9 shows, this tool can be used to check a domain controller's replication neighbors—a way of checking on your environment's replication topology. You'll also see if any replication attempts with those neighbors have succeeded or failed.

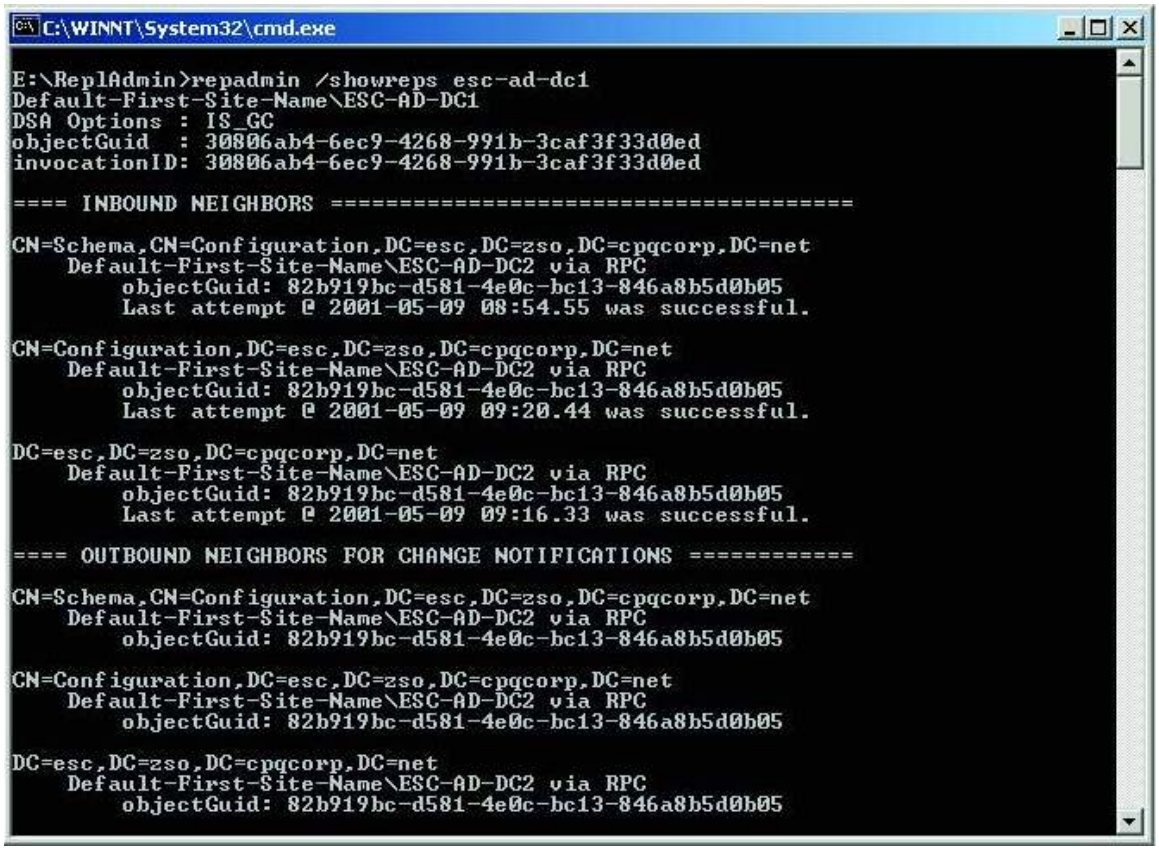

Figure 2.9: Using Repadmin to check replication status.

This—and other command-line tools—are great for checking real-time status information. What they're not good at is collecting information over the long haul, or for running continuously and proactively alerting you to problems.

Network Monitor

You might not ordinarily think of Network Monitor—or any packet-capture tool, including Wireshark and others—as a way of monitoring AD. In fact, with a lot of practice, they can be *great* tools. After all, much of what AD does ultimately comes down to network communications, and with a packet capture tool, you can easily see *exactly* what's transpiring over the network. Figure 2.10 illustrates the main difficulty in using these tools.

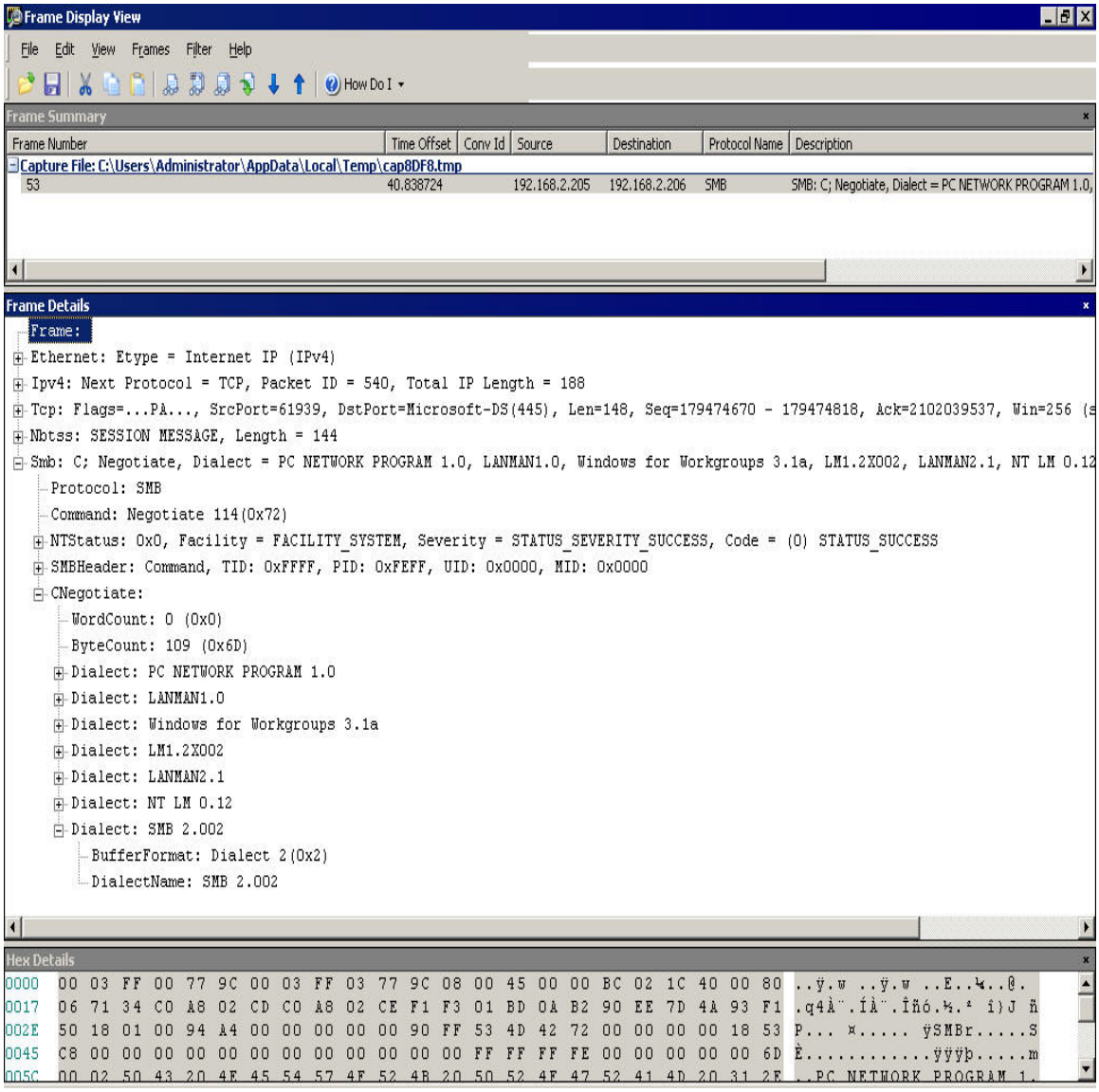

Figure 2.10: Captured AD traffic in Network Monitor.

You see the problem, right? This is rocket science-level stuff. I'm showing a captured packet for directory services traffic, but unless you know what this traffic *should* look like, it's impossible to tell whether this represents a problem. But gaining that knowledge is worth the time: I've used tools like this to find problems with DNS, Kerberos, time sync, and numerous other AD-related issues. Unfortunately, a complete discussion of these protocols, how they work, and what they should look like is far beyond the scope of this book.

At a simpler level, though, you can use packet capture tools as a kind of low-level workload monitor. For example, consider Figure 2.11.

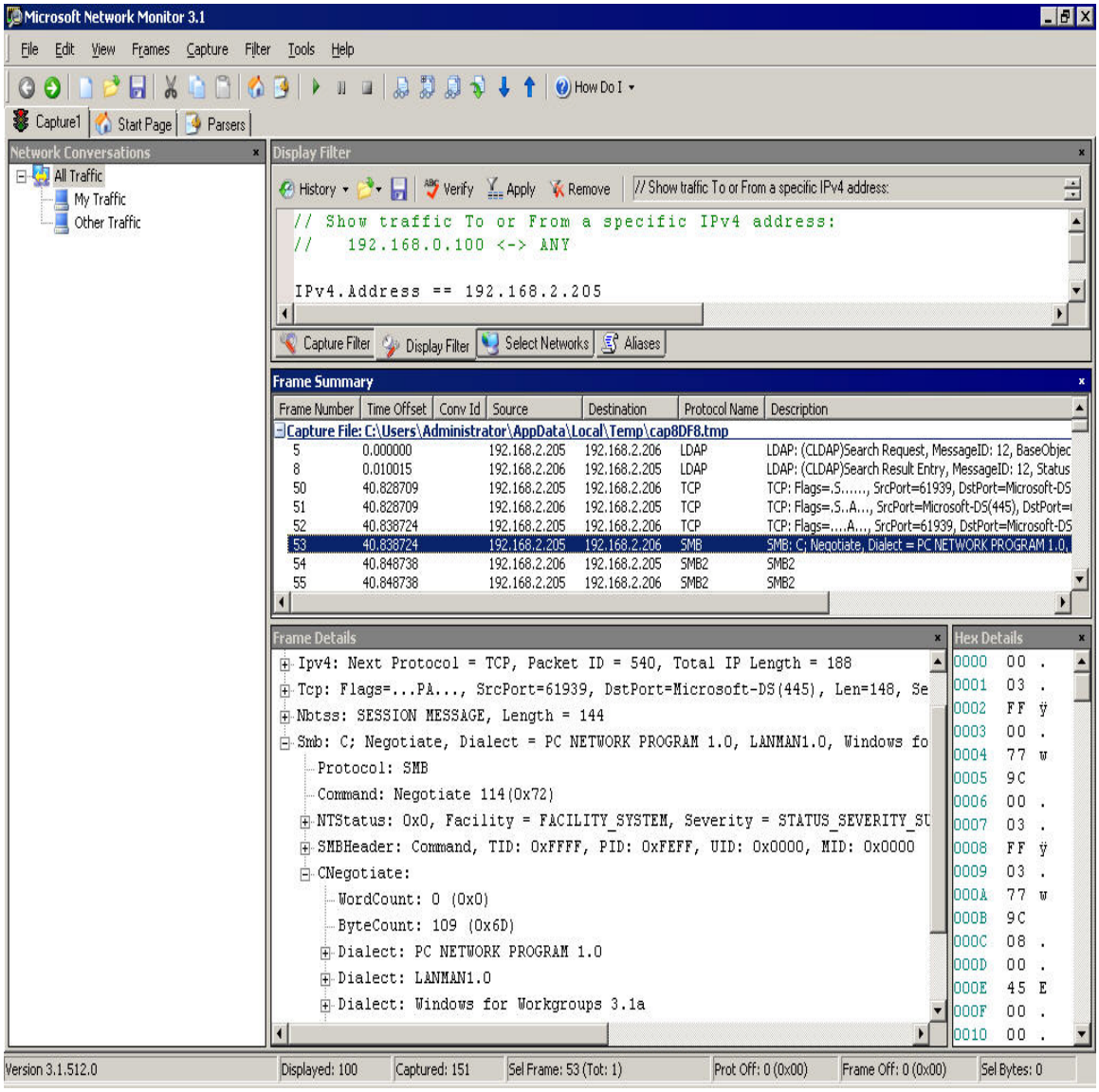

Figure 2.11: Capturing traffic in Network Monitor.

Ignoring the details of the protocol, pay attention to the middle frame. At the top of the packet list, you can see a few LDAP search packets. This gives me an idea of what kind of workload the domain controller is receiving, where it's coming from, and so forth. If I know a domain controller is overloaded, this can be the start of the process to discover where the workload is originating—in this case, it might be a new application submitting poorly-constructed LDAP queries to the directory.

System Center Operations Manager

System Center Operations Manager is Microsoft's commercial offering for monitoring both performance and functionality in AD as well as in numerous other Microsoft products and Windows subsystems. SCOM, as it's affectionately known, utilizes both performance counters and other data feeds much as Windows' native tools do. What sets SCOM apart are two things:

- Data is stored for a long period of time, enabling trending and other historical tasks
- Data is compared with a set of Microsoft-provided thresholds, packaged into Management Packs, that tell you when data represents a good, bad, or "going bad" condition

That last bit enables SCOM to more proactively alert you to performance conditions that are trending bad, and to then show you detailed real-time *and* historical data to help troubleshoot the problem. In many cases, Management Packs can include prescriptive advice for failure conditions, helping you to troubleshoot and solve problems more rapidly. As a tool, SCOM addresses most, if not all, of the weaknesses in the native Windows toolset. It does so by relying primarily on native technologies, and it does so in a way that often imposes less monitoring overhead than some of the native tools. Having SCOM collect performance data for a month, for example, is a lot easier on the monitored server than running Performance Monitor continuously on that server. SCOM does, however, require its own infrastructure of servers and other dependencies, so it adds some complexity to your environment.

Unfortunately, one of SCOM's greatest strengths—its ability to monitor a wide variety of products and technologies from a single console—is also a kind of weakness because it doesn't offer a lot of technology-specific functionality. For example, SCOM isn't a great way to construct an AD replication topology map because that's a very AD-specific capability that wouldn't be used by any other product. In other words, SCOM is a bit generic. Although it can provide great information, and good prescriptive advice, it isn't necessarily the *only* tool you'll need to troubleshoot every problem. SCOM can alert you to most types of problems (such as an unacceptably high number of replication failures), but it can't always help you visualize the underlying data in the most helpful way.

Third-Party Tools to Consider

I'm not normally a fan of pitching third-party products, and I'm not really going to do so here. That said, we've identified some weaknesses in the native tools provided with Windows. Some of those weaknesses are addressed by SCOM, but because that tool itself is a commercial add-on (that is, it doesn't come free with Windows), you owe it to yourself to consider *other* add-on commercial tools that might address the native tools' weaknesses in other ways, or perhaps at a different price point. That said, what are some of the weaknesses that we're trying to address?

Weaknesses of the Native Tools

Although I think Microsoft has provided some great underlying technologies in things like event logs and performance counters, the tools they provide to work with those are pretty basic. In order to decide if a replacement tool is suitable, we need to see if it can correct these weaknesses:

- Non-centralized—Windows' tools are per-server, and when you're talking about AD, you're talking about an inherently distributed system than functions as a single, complicated unit. We need tools that can bring diagnostic and performance information together into a single place.
- Raw data—Windows' tools really just provide GUI access to underlying raw data, either in the form of events or performance counters or whatever. That's really sub-optimal. What we want is something to translate that data into English, tell us what it means, and possibly provide intelligence around it—which is a lot of what SCOM offers, really.
- Limited data—Windows' tools collect the information available to them through native diagnostic and performance technologies—and that's it. There are certainly instances when we might want *more* data, especially more-specific data that deals with AD and its unique issues.
- Generic—Windows' tools are pretty generic. The Event Viewer and Performance Monitor, for example, aren't AD-specific. But an AD-specific tool could go a long way in making both monitoring and troubleshooting easier because it could present information in a very AD-centric fashion.

Ways to Address Native Weaknesses

There are a few ways that vendors work to address these weaknesses:

- Centralization—Bringing data together into one place is almost the first thing any vendor seeks to address when building a toolset. Even Microsoft did this with SCOM.
- Intelligence—Translating raw data into processed information—telling us if something is "good" or "bad," for example—is one way a tool can add a great deal of value. Prescriptive advice, such as providing advice on what a particular event ID means and what to do about it, is also useful. This kind of built-in "knowledge base" is a major selling point for some tool sets.
- More data—Some tools either supplement or bypass the native data stores and collect more-detailed data straight from the source. This might involve tapping into LDAP APIs, AD's internal APIs, and so forth.
- Task-specific—Tools that are specifically designed to address AD monitoring can often do so in a much more helpful way than a generic tool can. Replication topology maps, data flow dashboards, and so forth all help us focus on AD's specific issues.

Vendors in this Space

There are a *lot* of players in this space. A lot a lot. Some of the major names include:

- Quest
- ManageEngine
- Microsoft
- Blackbird Management Group
- NetIQ
- IBM
- NetPro (which was purchased by Quest)

Most of these vendors offer tools that address native weaknesses in a variety of ways. Some utilize underlying native technologies (event logs, performance counters, and so forth) but gather, store, and present the data in different ways. Others bypass these native technologies entirely, instead plugging directly into AD's internals to gather a greater amount of information, different information, and so forth.

In addition, there are a number of smaller tools out there that have been produced by the broader IT community and smaller vendors. A search engine is a good way to identify these, especially if you have specific keywords (like "replication troubleshooting") that you can punch into that search engine.

Let's Start Troubleshooting

Now that you know how to keep an eye on what AD is doing, you're ready to dive into troubleshooting the directory when it isn't doing the right thing. In the next chapter, I'll introduce you to a structured directory troubleshooting approach developed by Directory Services MVP Award recipient Sean Deuby. We'll use Sean's approach as a guide toward tracking down problematic AD subsystems and solving problems. At the same time, I'll be explaining core troubleshooting techniques that will help make you a more efficient and effective troubleshooter.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.