Create graphical reports for Exchange 2007 - Part 1

In this series we will show you how to create eye-catching graphical reports from software for Exchange Server.

Network Management - In this series we will show you how to create eye-catching graphical reports from software for Exchange Server.

Introduce

One of the most common requirements from all correspondent administrators around the world is the ability to create comprehensive reports for the environment they manage. If a picture is worth thousands of words then people can easily understand the need to convert thousands of log files that an Exchange server can create into something human-friendly.

Since Exchange Server 2007 (and all previous versions) is unable to create rich graphical reports, there are a few companies that have filled this gap by building reporting software. for Microsoft Exchange.

With Microsoft, the software giant's 2005 release, SQL Server 2000 Report Pack for Microsoft Exchange, is a separate version of Exchange Reporter, a commercial software developed by the company and called SSW. In fact, very few people have used SQL Server 2000 Report Pack for Microsoft Exchange because the requirements of SQL Server and SQL Server Reporting Services are quite high.

Microsoft has taken quite an interesting move with another method. Microsoft Operations Manager (MOM) 2005 with Exchange Server Management Pack has many powerful report generation features. However, Exchange Server Management Pack for the next version of Microsoft management software, System Center Operations Manager (SCOM) 2007 has lost some previous reports still available. However, Microsoft Exchange Server 2007 Management Pack for the latest System Center Operations Manager 2007 R2 provides more than 30 specific reports for Exchange Server 2007 to test availability and performance.

When we decided to introduce this tutorial, we never had the idea of ??using commercial software, or using the reporting features of system management software but instead we wanted to provide gives you the ability to graphically report to Exchange Server, using a simple and free tool, some from Microsoft, and others developed by Microsoft MVP and some individuals all over the world gender.

All are in the log files

Above I have explained that Exchange Server does not have native graphic reporting features, but this does not mean that it cannot generate the information you need (even more). All are in log files!

What kind of information log files are there in the regular Exchange Server organization? Table 1 below lists the most common records. Note that these logs can be spread across all Exchange servers, because all Exchange roles cannot be located on the same machine.

Record

Default path

Protocol Logs (SMTP Send)

Exchange ServerTransportRolesLogsProtocolLogSmtpSend

Protocol Logs (SMTP Receive)

Exchange ServerTransportRolesLogsProtocolLogSmtpReceive

Agent Logs

Exchange ServerTransportRolesLogsAgentLog

IIS Logs

[Windows 2003] WindowsSystem32LogFilesW3SVC1
[Windows 2008] InetpubLogsLogFilesW3SVC1

Message Tracking Logs

Exchange ServerTransportRolesLogsMessageTracking

POP3 / IMAP Logs

Exchange ServerClientAccessPopImap

Connectivity Logs

Exchange ServerTransportRolesLogsConnectivity

Pipeline Tracing Logs

Exchange ServerTransport RolesLogsPipelineTracing

Routing Table Logs

Exchange ServerTransportRolesLogsRouting

MRM Logs

Exchange ServerLoggingManaged Folder Assistant

Table 1: General log files in Exchange

The next step is to properly configure the logging, since not all records are enabled by default and one of them needs some adjustments to the calendar data we want to keep.

In this section, we will only use 5 log files from the table above. In these 5 records, there are two records that are not enabled by default: protocol logs: SMTP Send and SMTP Receive. The SMTP transmission logging level is

controlled at the Exchange connector level.

To create useful Exchange graphics reports, we assume that the protocol logs from servers on the network boundary are very important, because they register SMTP transaction sessions from incoming or outgoing mail. your organization. In case you deployed the Edge transport server, these logs can be configured from the internal Hub Transport server (if you modify the connector properties from the Edge server, you will get an error like shown in Figure 1).

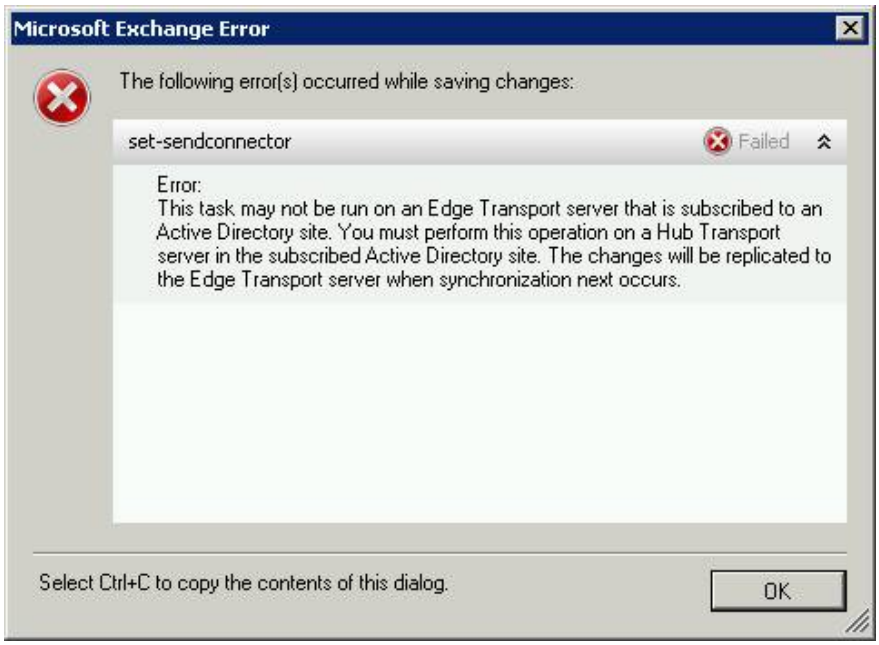

Figure 1: Error while changing the level when on the Edge server

To enable SMTP Protocol Logs of EdgeSync Send Connectors, open the Exchange Management Console, open **Organization Configuration** , select **Hub Transport,** then on the right pane, click the **Send Connectors** tab. Right-click on the two connectors and select **Properties** (Figure 2).

On the EdgeSync Connector Properties window, change the **Protocol logging level** to **Verbose** (Figure 3).

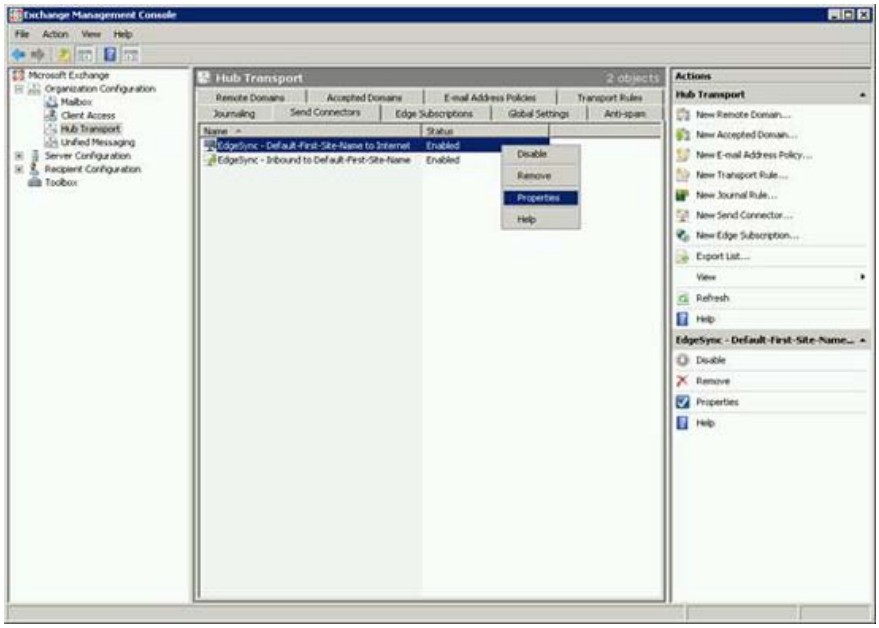

Figure 2: Configuring EdgeSync Send Connector

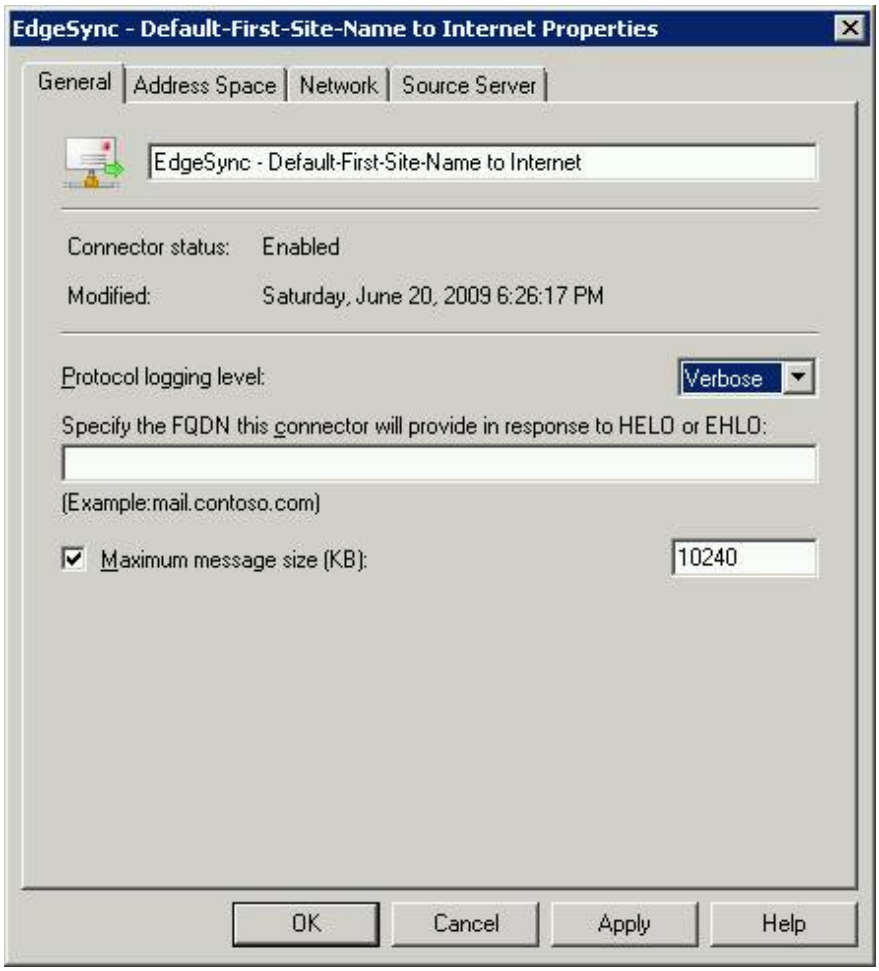

Figure 3: EdgeSync Send Connector properties

If you prefer to use PowerShell, run the following command (for both connectors) to set the write level to Verbose (Figure 4):

Set-SendConnector "EdgeSync - Inbound to Default-First-Site-Name" -ProtocolLoggingLevel Verbose

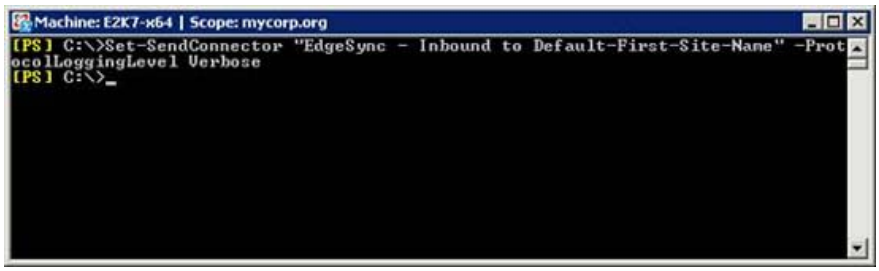

Figure 4: Change the write level with PowerShell

After enabling SMTP Transport Logs, we must define the amount of historical data to keep. Exchange Server allows us to control the maximum log file size, maximum directory size and maximum log file life using the *Set-TransportServer* command in PowerShell.

The *SendProtocolLogMaxDirectorySize* and *ReceiveProtocolLogMaxDirectorySize* parameters specify the maximum size of the Send folder and the Receive Connector Protocol Log. When the maximum folder size is reached, the server will delete the oldest log files. The minimum value is 1MB, the default value is set to 250MB.

Since the default size of 250MB is not enough, let's change the maximum size of the Send Connector folder to 2GB and the Receive connector to 4GB using the Exchange Management Shell command:

Set-TransportServer -Identity E2K7EDGE -SendProtocolLogMaxDirectorySize 2048MB -ReceiveProtocolLogMaxDirectorySize 4096MB

Now that we are ready for our records, this is the time to start analyzing them.

Be aware that, depending on the quality of the data you are analyzing, parsing and processing the logs may take place quickly or slowly.

Parsing logs

Log Parser log parser is a powerful tool, which allows you to query text data such as log files, XML files, CSV files as well as the main data source on the operating system. Windows, Event Log, Registry, file system or even Active Directory. In addition to providing parsing information, Log Parser also results in queries in custom format at the output, such as data datagrid grid, or can be converted into visual charts. .

Log Parser does not need to be installed on Exchange Server, all you need to do is ensure access for Exchange log directories.

Follow the instructions below to install Microsoft Log Parser:

1. Download and install Microsoft Logparser 2.2.
2. Download and install Office 2003 Add-in: Office Web Components. This is the necessary operation to provide graphical features for Log Parser.
3. Download and install Microsoft Office 2003 Web Components Service Pack 1 (SP1) for the 2007 Microsoft Office System.

Log Parser has a fairly complete help file (Figure 5), by default located at C: Program Files (x86) Log Parser 2.2, you should read this help carefully. There are also a few examples provided at C: Program Files (x86) Log Parser 2.2Samples for your reference.

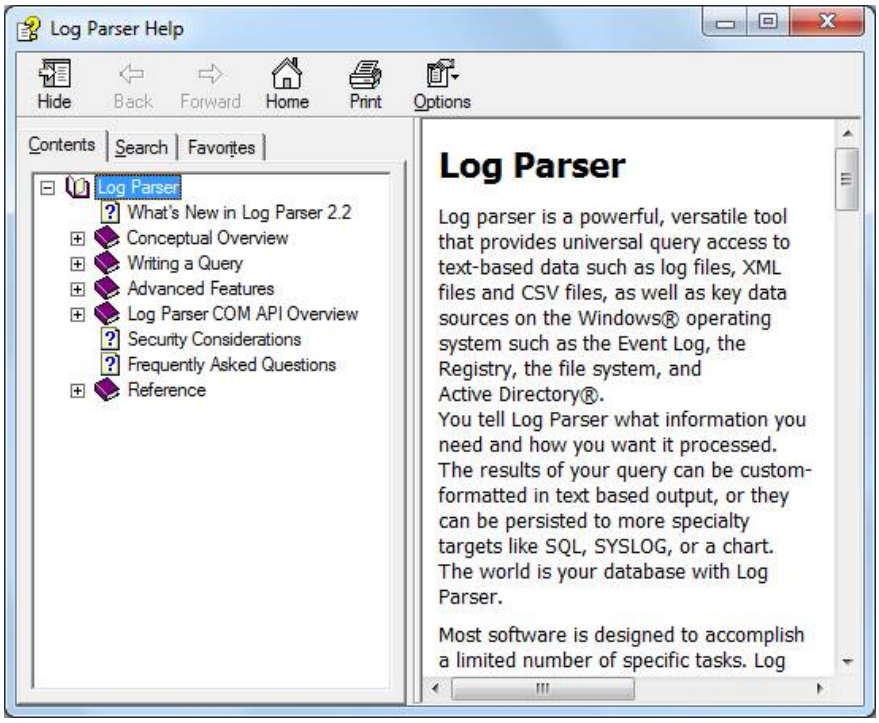

Figure 5: Log Parser file

In the following sections, we will show you some examples of Log Parser queries used to create the desired visual reports. These queries can be run directly from the command line or you can create batch files different from each query (should be done).

Log Parser usage reports with Agent logs

If you are using Exchange Server's anti-spam agents, there are a few reports we can get by analyzing the Agent logs. These logs are on the Exchange Edge server, if you are using it, or in an Exchange Hub server, in case it has enabled anti-spam agents and is running.

To get the idea of ??mail coming into your organization, we can start by organizing the number of messages according to their Spam Confidence Level (SCL) and displaying them in a datagrid format.

This is the order to create that datagrid (Figure 6):

"C: Program Files (x86) Log Parser 2.2logparser.exe" "SELECT ReasonData, count (*) AS hits FROM C: Progra ~ 1MicrosoftExchan ~ 1TransportRolesLogsAgentLogAGENT * .log WHERE ReasonData > NULL GROUP BY ReasonData ORDER BY DESC hits" - i: CSV -nSkipLines: 4 -o: DATAGRID -dtlines: 800 -rtp: -1

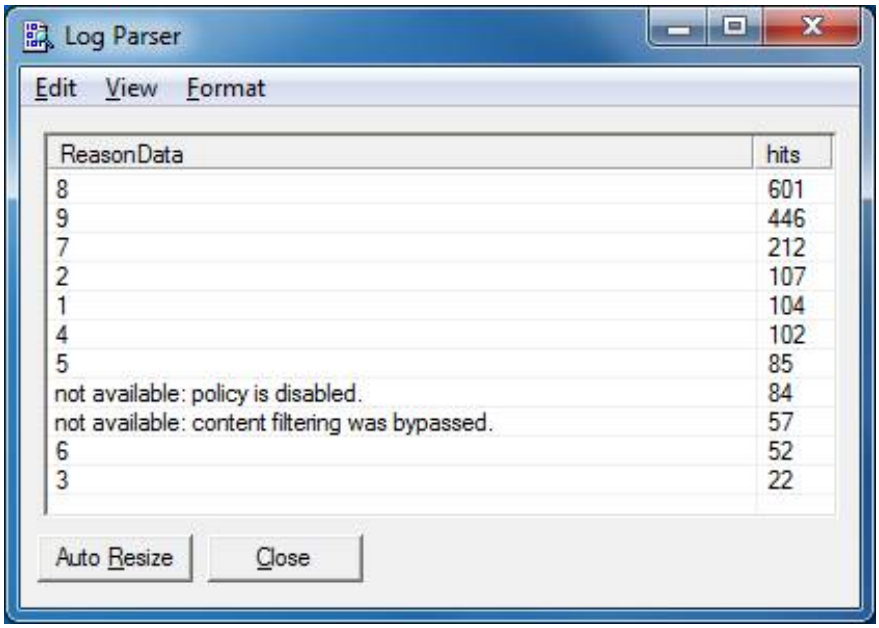

Figure 6: Agent reason spread (Datagrid)

If you like a chart for previous results, it is easy to do. By using the parameter *-chartType: PieExplode3D* in the command below, we will get a visual diagram as shown in Figure 7.

"C: Program Files (x86) Log Parser 2.2logparser.exe" "SELECT CASE TO_INT (ReasonData) WHEN NULL THEN 0 ELSE TO_INT (ReasonData) END AS ReasonData2, count (*) AS hits INTO agentreasonspread.gif from C: Progra ~ 1MicrosoftExchan ~ 1TransportRolesLogsAgentLogAGENT * .log GROUP BY ReasonData2 ORDER BY DESC hits "-i: CSV -nSkipLines: 4 -o: CHART -chartType: PieExploded3D -chartTitle:" Agent Reason Spread "-e 200 -dtlines: 600

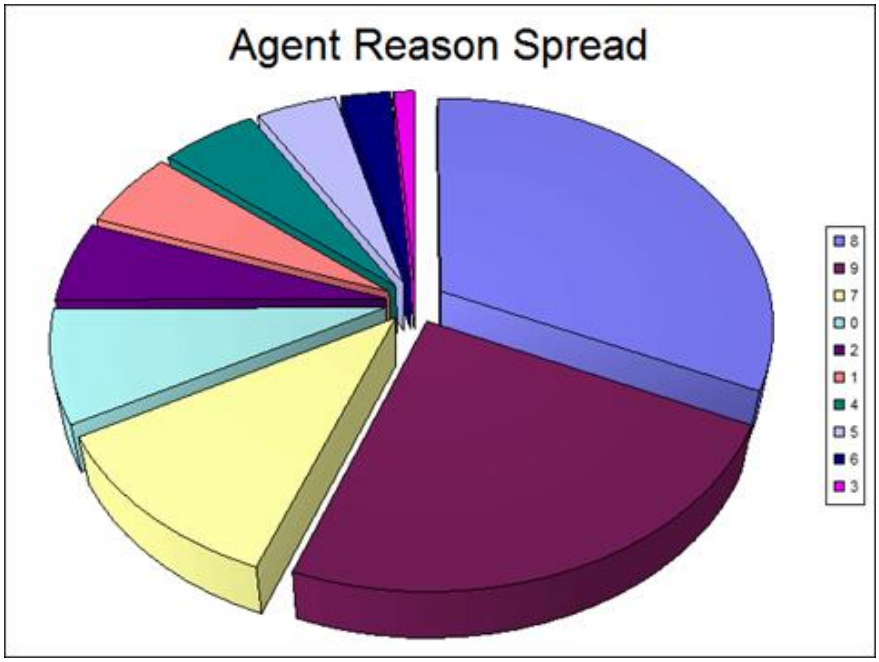

Figure 7: Agent reason spread

Although SCL changes from 1 to 9, but you will see that there is a thin slice of the previous chart with a value of 0. The value 0 represents all disabled policies and content filtering functionality disable (see previous datagrid), meaning that it shows mail coming into your organization.

If you prefer to have a more integrated view of the previous chart, with only accepted mail and removed, the following logparser query will do that. Note that messages with an SCL ratio of 8 or higher will be considered removed, SCL 7 means quarantined, the rest is accepted.

"C: Program Files (x86) Log Parser 2.2logparser.exe" "SELECT CASE TO_INT (ReasonData) WHEN 9 THEN 'REJECTED' WHEN 8 THEN 'REJECTED' WHEN 7 THEN 'QUARANTINED' ELSE 'ACCEPTED' END AS ReasonData2, TO_INT (mul (100.0, PropCount (*))) as Percent, count (*) as hits INTO agentAcceptedRejected.gif FROM C: Progra ~ 1MicrosoftExchan ~ 1TransportRolesLogsAgentLogAGENT * .log GROUP BY ReasonData2 ORDER BY hits DESC "-i: CSV -nSkipLines: 4 -o: CHART -chartType: PieExploded3D -chartTitle: "%% Accepted / Rejected mail" -dtlines: 600 -categories: OFF -values: ON -view: ON

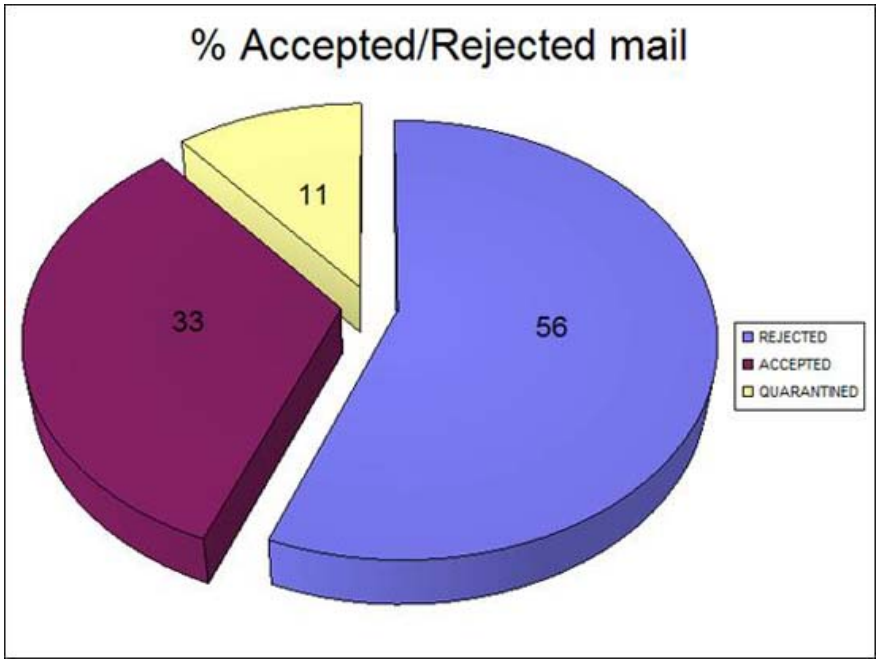

Figure 8:% mail is accepted and removed

Reports with Log Parser with protocol logs

In the next section, we will use SMTP protocol logs. With these protocol logs, we can extract useful information about the partitions of SMTP connections and about hosts (not users).

To get an image of Total Inbound Simultaneous Connections, we will use the code below:

"C: Program Files (x86)" 2.2logparser.exe Loger "" SELECT QUANTIZE (TO_TIMESTAMP (EXTRACT_PREFIX (TO_STRING (EXTRACT_SUFFIX ([# Fields: date-time], 0, 'T')), 0, '.') , 'hh: mm: ss'), 3600) AS Hour, COUNT (*) AS Hits INTO radar_traffic.gif FROM C: Progra ~ 1MicrosoftExchan ~ 1TransportRolesLogsProtocolLogSmtpReceiveRECV * .LOG WHERE event = '+' GROUP BY Hour ORDER BY Hour ASC " -i: CSV -nSkipLines: 4 -o: CHART -charttype: RadarLineFilled -charttitle: "Global total SMTP inbound simultaneous connections per hour"

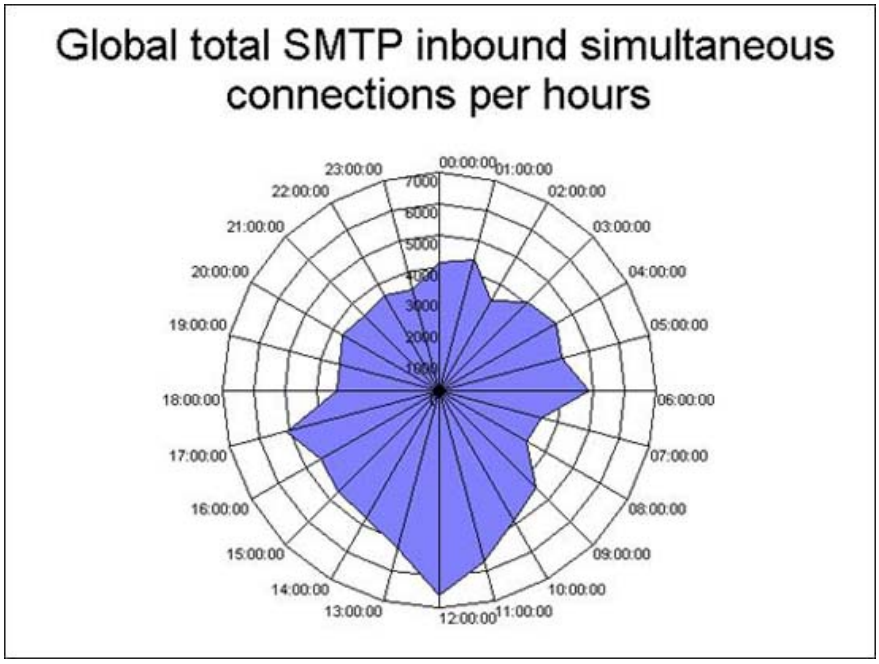

Figure 9: SMTP connections enter

If you like Figure 9 and you like to see a similar chart for outbound connections, the output of the command below is a radar chart described in Figure 10.

"C: Program Files (x86)" 2.2logparser.exe Loger "" SELECT QUANTIZE (TO_TIMESTAMP (EXTRACT_PREFIX (TO_STRING (EXTRACT_SUFFIX ([# Fields: date-time], 0, 'T')), 0, '.') , 'hh: mm: ss'), 3600) AS Hour, COUNT (*) AS Hits INTO radar_traffic_send.gif FROM C: Progra ~ 1MicrosoftExchan ~ 1TransportRolesLogsProtocolLogSmtpSendSEND * .LOG WHERE event = '+' GROUP BY Hour ORDER BY Hour ASC " -i: CSV -nSkipLines: 4 -o: CHART -charttype: RadarLineFilled -charttitle: "Global total SMTP outbound simultaneous connections per hour"

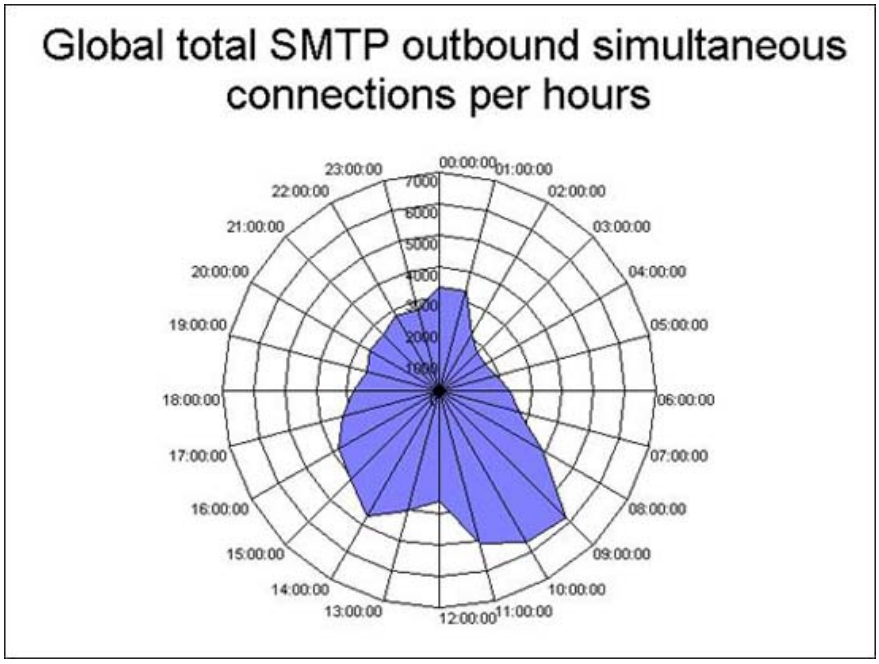

Figure 10: Connections simultaneously come out

The next order analyzes what is the suspect sender to your organization. To accomplish that goal, we need to extract from SMTP Receive Log all hosts that have status codes of 500 and larger, such as 504, 535, 550, .

This process is done in two steps: the first logparser query extracts data from the logs, the second command performs a reverse DNS lookup for the original output. The reason we need to follow these two steps is to take a time-consuming reverse DNS lookup process when extracting data.

"C: Program Files (x86) Log Parser 2.2logparser.exe" "SELECT EXTRACT_PREFIX (remote-endpoint, 0, ':') AS Remote-host, count (*) AS hits INTO SuspiciousSenders.xml FROM C: Progra ~ 1MicrosoftExchan ~ 1TransportRolesLogsProtocolLogSmtpReceiveRECV * .log WHERE TO_INT (SUBSTR (DATA, 0.3))> 500 AND event = '>' GROUP BY Remote-host ORDER BY DESC hits "-i: CSV -nSkipLines: 4 -o: XML
"C: Program Files (x86) Log Parser 2.2logparser.exe" "SELECT TOP 10 REVERSEDNS (Remote-host), hits FROM SuspiciousSenders.xml" -i: XML -o: DATAGRID

Note that there is an internal host on the results described in Figure 11. This host may be an internal application server or an authenticated internal mail forwarding.

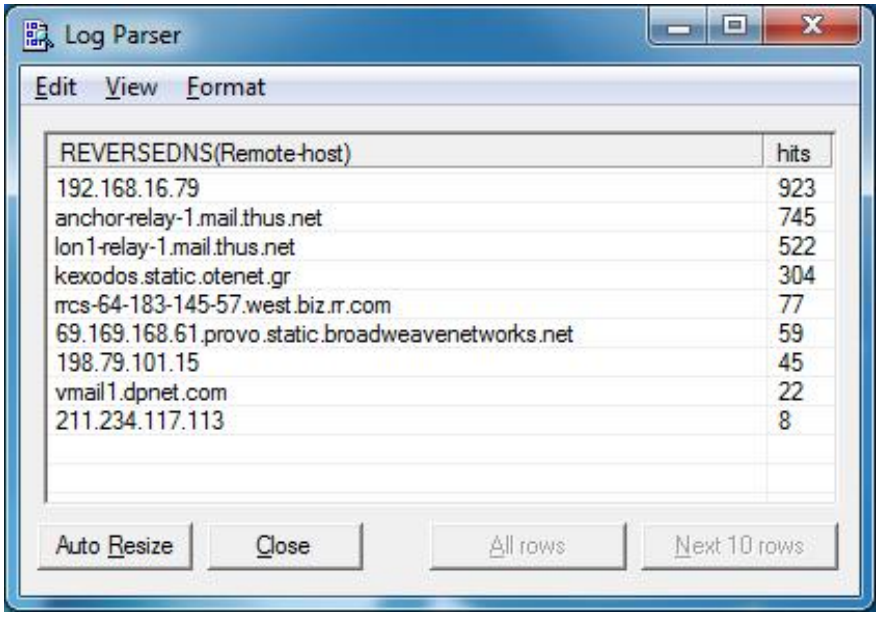

Figure 11: Suspicious hosts are sending mail to your organization

We can also create a datagrid with the Top Outbound Rejection Errors by analyzing the SMTP Send Protocol Log. This is useful for identifying outbound errors or in finding whether your server is listed in some blacklist. Here is the command to create the datagrid from Figure 12:

"C: Program Files (x86) Log Parser 2.2logparser.exe" "SELECT CASE TO_INT (SUBSTR (DATA, 0.3)) when NULL then 0 else TO_INT (SUBSTR (DATA, 0.3)) END AS RemoteHostReturnCode, data , count (*) AS hits FROM C: Progra ~ 1MicrosoftExchan ~ 1TransportRolesLogsProtocolLogSmtpSendSEND * .log WHERE RemoteHostReturnCode> 400 AND context > 'Certificate' : CSV -nSkipLines: 4 -o: DATAGRID

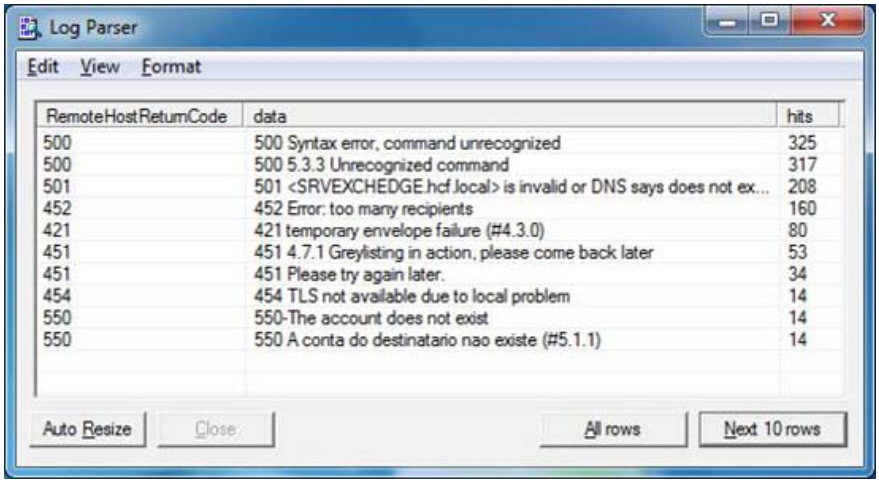

Figure 12: Top rejection errors sent

One of the most desirable reports is the distinction between top senders and organizations. The answer is bundled in these SMTP Receive Transport Logs logs.

Note in the two-step process above, reverse DNS lookup is only performed with the output from the first query, the purpose to optimize the time it takes place.

"C: Program Files (x86) Log Parser 2.2logparser.exe" "SELECT TOP 10 EXTRACT_PREFIX (remote-endpoint, 0, ':') AS RemoteSendingHost, count (*) AS Hits INTO topsenders.xml FROM C: Progra ~ 1MicrosoftExchan ~ 1TransportRolesLogsProtocolLogSmtpReceiveRECV * .LOG WHERE event = '+' GROUP BY RemoteSendingHost ORDER BY Hits DESC "-i: CSV -nSkipLines: 4 -o: XML
"C: Program Files (x86) Log Parser 2.2logparser.exe" "SELECT TOP 10 REVERSEDNS (RemoteSendinghost), Hits INTO topsenders.gif FROM TopSenders.xml" -i: XML -o: CHART -chartType: PieExploded3D -chartTitle: " TOP 10 Senders "-groupSize: 1024x768

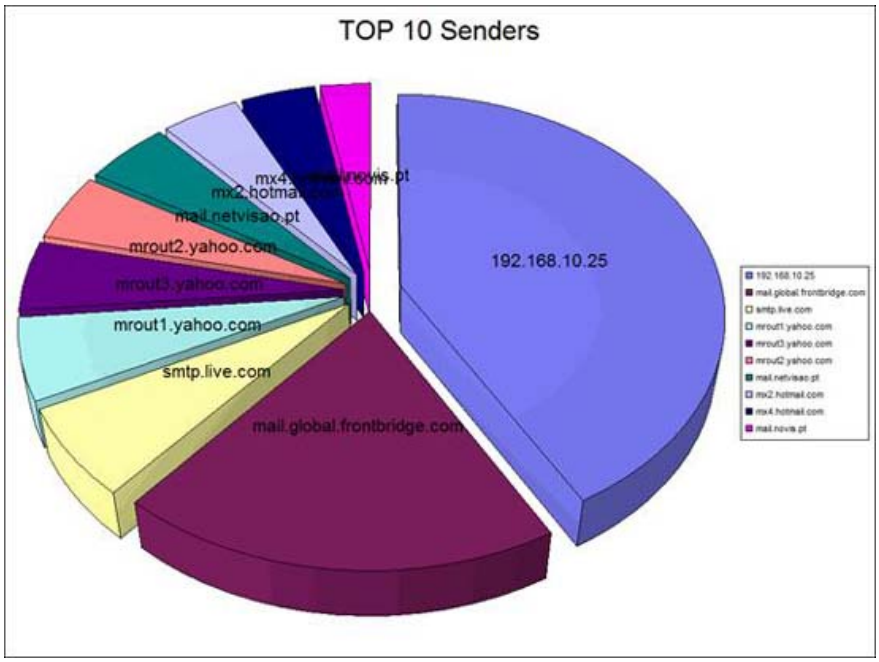

Figure 13: Top sender hosts into the organization

Conclusion

You might be wondering why so many Exchange logs are needed, or you think you might consider unwanted log files, but the work is very useful and is the key to opening all. Your type of information about your mail infrastructure. In the next part of this series, I will show you some Log Parser related issues and prepare for some other queries!