Global Information Assurance Certification Paper

Centralizing Event Logs on Windows 2000

By Gregory Lalla

February 28, 2003

GIAC Security Essentials Certification Practical Assignment
Version 1.4b

Brief

This case study will detail how I setup a central repository for server logs and daily notifications of events that might indicate a security incident. This was done on a limited budget using free tools available from the internet and software already in use for other projects. My goal was to consolidate the Eventviewer logs, Internet Information Services (IIS) logs, and Urlscan logs from 15 Windows 2000 web servers into a database I could query against. I would then have the results of the queries automatically emailed to me.

The consolidation of the logs to a central server turned out to be a fairly easy task using Dumpevt, located at http://www.systemtools.com/somarsoft/, for the Eventviewer logs and a simple batch file to copy the IIS and Urlscan logs. Getting the data into and then out of a database, however, proved to be extremely difficult and time consuming. I used a tool called Log Parser, located at http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=8CDE4028-E247-45BE-BAB9-AC851FC166A4, to import the IIS and Urlscan logs into the database and a SQL Server Agent Job, which used the BCP command to import the dump files of the Eventviewer logs into the database. I then wrote SQL queries in a SQL Server Agent Job and used a tool called Blat to email me the results. The completion of this project enhanced dramatically the overall security of my servers. I can easily see the activities that most effect me without having to check hundreds of logs and filtering through thousands of irrelevant entries without missing events that might allow an attacker to go undetected.

Before

The need to centrally locate my event logs was the result of a couple of factors. First and for most, I wanted a convenient way to look at and manage my logs. Secondly, there was a security incident that happened in our organization where several UNIX servers where hacked into. After an investigation into the incident, a recommendation was issued stating: “Configure system logging to provide a higher level of detail, and maintain logs for more than 30 days…review system logs on a routine basis, such as daily…use a centralized log host for multiple systems. Each system should log a copy of their local logs to the centralized log host. This will simplify consolidated log reviews and ensure a second copy of the logs is available in case of a problem with the system.” Our department decided to make this our policy with a log retention of 60 days.

I manage 15 Windows 2000 web servers with over 100 web sites. Many of these web sites are open to the world. These servers are set to get their time from a Time Server on the network. I do not have any control over the network or the firewall. My organization is a high profile target and my servers are constantly bombarded with the Nimda, Code Red, probes and other malicious activities. I followed the Sans Securing Windows 2000 Step by Step Guide and have enabled auditing on all my servers. I audit the success and failure of Account

Logon Events, Account Management, Logon Events, Object Access, Policy Change, Privilege Use and System Events. As a result, I have substantial size log files that make it nearly impossible for a single individual to go through them each day looking for suspicious activities. The potential of missing such activities and allowing a hacker to go undetected is enormous.

Our organization also has a very limited budget and we were not able to purchase any of the commercially available logging software like Microsoft's MOM or NetIQ's Security Manger. I needed to find a solution to our problem without spending much money. At my disposal were a Microsoft MSQL 2000 server, Microsoft Access software, and the internet. Unfortunately, I have very limited knowledge about databases and SQL statements. My role in maintaining the MSQL database server is to keep it patched and to help the Database Administrators (DBA) create DOS scripts which import data from a mainframe into the databases.

During

I started by outlining what I wanted to accomplish. The requirements were that the log files would automatically be moved each night off of each server onto a centralized server and deleted from the original location. I needed a backup taken during our nightly backups of each log file before they were deleted from the servers. I wanted the logs files to automatically be imported into the database that same night and to clear those logs for the next night's imports. After the data was in the database, I wanted queries to run against it to extract the information that I was concerned about. I also wanted to have the results of those queries emailed to me in the morning, before I arrived at work, in a format that was easy to read.

I went through my SANS Security Essentials course and found several tools in the Windows Audit sections that dealt with log file management. These utilities were Dumpel located at, http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/dumpel-o.asp, NTLast locate at, http://www.foundstone.com/knowledge/proddesc/ntlast.html, Dumpevt, and NTOlog which is now apart of 'Fire and Water' located at, http://www.ntobjectives.com/products/firewater/. I then did a search of Windows event log tools on www.google.com and came across several websites and articles that mentioned different utilities. One article was, "Archiving and Analyzing the NT Security Log" by Randy Smith, Windows & Net Magazine located at http://www.win2000mag.com/Articles/Index.cfm?ArticleID=9043. This article pointed out the tools EventSave located at, http://www.heysoft.de/Frames/f_sw_es_en.htm and ELDump located at http://www.ibt.ku.dk/jesper/eldump/eldump.htm. I found out about the tool Log Parser from the IISFAQ web site located at, http://www.iisfaq.com/default.aspx?View=A525&P=1. I found out about the tool

Log Agent from a paper by Floydman called, "Log Agent, Log File Recollection Tool" located at, http://iquebec.ifrance.com/securit/whitepapers/logagent.html. And I found reference to the EventcombMT tool in the Security Operations Guide for Windows 2000 Server located at, http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/windows/windows2000/staysecure/default.asp.

I downloaded all these tools along with their documentation. After reading about and testing each tool, I quickly eliminated the tools that could not do any of the tasks listed in my requirements documentation. This included, NTLast and EventcombMT, which only searched on the log files themselves and could not be used against the database. I also eliminated Log Agent. This tool does centralize logs but not in the way I needed. Log Agent takes logs generated by programs and sends them to a central location where they are appended with any changes. This would work well for applications that produce a single log but IIS and Urlscan logs are created new each day, which makes it simpler to just copy the logs over with a batch file to a central location. Log Agent also cannot produce an Eventviewer log in a format that's easily imported into a database, like a comma delimited file.

I took a closer look at the remaining tools. Dumpel allowed me to remotely import the event logs from multiple servers in a comma delimited file but it had one limitation. It did not clear the log file after the Eventviewer logs were dumped. I wanted to have the logs cleared for a couple of reasons. First, I didn't want any events to be duplicated in the database. With the logs cleared, this could not happen. Second, for security reasons, I did not want to have past logs available to a hacker who broke in to the system. They might be able to gather further information which could result in more harm. This was also the flaw with Log Parser and ELDump, though you could use another utility called ElSavClr located at, http://www.ibt.ku.dk/jesper/elsavClr/, to clear the files after the dump occurred. I also did not want to have to set up two products when one would do. Plus, there is a possibility of erasing some events that were not dumped between the time you dump the log and the time you clear the log. NTOlog produced a backup of the Eventviewer logs but you couldn't get the output in a format that was easily imported into a database. It also did not automatically copy the files it did backup over the network to the central server. EventSave had a couple of limitations. It cleared the Eventviewer logs but you had to install the program on each server (unless you wanted the logs from every server in the domain) and then point the output to a central server. It also could not output the logs in a comma delimited format.

The tool that seemed to fit my criteria was Dumpevt. With a simple command, I could choose a remote server and have it dump its Eventviewer logs onto the central server and then have the logs cleared. I decided that this would be the tool to use for the Eventviewer logs. Now I had to figure out how to get the IIS

and Urlscan logs to the central server. A batch file to copy over the log files seemed like the easiest way.

On the central server, which is called KODOS, I copied over the Dumpevt utility and placed it in the directory D:\Eventlog\Dumpevt. I also created a directory called Output, underneath Dumpevt for the dumped Eventviewer logs. I then wrote a batch file called evtdump.bat, which would centralize the logs from all the servers onto KODOS and then clear the logs. The following is the syntax I used in the batch file for each server:

```
Dumpevt.exe /computer=[servername] /logfile=sys
/outfile=d:\eventlog\dumpevt\output\[servername]sys.txt /clear

Dumpevt.exe /computer=[servername] /logfile=app
/outfile=d:\eventlog\dumpevt\output\[servername]app.txt /clear

Dumpevt.exe /computer=[servername] /logfile=sec
/outfile=d:\eventlog\dumpevt\output\[servername]sec.txt /clear
```

I scheduled the batch file to run at 5:45am every morning with a scheduled task. I chose this time because I wanted the backup to be able to run each night and backup the Eventviewer logs before I cleared them. In that case, if anything happened to the dumps, I would be able to go back and retrieve the data from a restore on the servers. This also allows the administrator to take a look in the logs for an entire day to check for any problems or errors that might need to be researched. If the administrator needs to go further back than a day, they can write a query and get the output from the database.

I also needed to create a directory structure on KODOS where the IIS and Urlscan logs were to be copied. Under D:\Eventlogs, I made a directory called Logs. Under Logs, I created a directory for each of the servers using the server name. Then under each of the server names I created a Urlscan directory and a W3SVC*[#]* directory where the # is the web site instance. On each server I wrote a batch file that included the following syntax:

```
net use I: \\kodos\d$
c:
cd c:\winnt\system32\logfiles\w3svc1
copy *.* I:\eventlog\logs\[servername]\w3svc1
cd c:\winnt\system32\logfiles\w3svc2
copy I:\eventlog\logs\[servername]\w3svc2
[etc]…
cd C:\WINNT\system32\inetsrv\urlscan
copy *.log I:\eventlog\logs\[servername]\urlscan
net use /delete I:
```

This batch file, called copylog.bat, created a mapping to the KODOS, changed to the C drive, navigated to the log files, copied them over to KODOS and then disconnected the mapping. For the Urlscan logs, I only copied the .log files over since there are other files in that directory that are not logs. I then wrote another batch file on each server called dellog.bat, which deleted all the IIS and Urlscan logs on that server after they were copied over to KODOS. I did this for the same reason the Eventviewer logs were cleared. The syntax is as follows:

```
c:
cd c:\winnt\system32\logfiles\w3svc1
del *.* /Q
cd c:\winnt\system32\logfiles\w3svc2
del *.* /Q
[etc]…
cd C:\WINNT\system32\inetsrv\urlscan
del *.log /Q
```

The /Q option tells the delete command to operate in Quite mode and not ask for confirmation on wildcards. I scheduled copylog.bat and dellog.bat to run each night with a scheduled task. Copylog.bat was scheduled to run at 12:00am. I chose this time because it is the time right before a new log is created for the following day. That way, I only get the previous days log and I get it in its entirety. The dellog.bat file is scheduled to run at 1:00am giving the copylog.bat enough time to complete before the files are deleted. It is also important to note that the current log file will not be deleted with the del *.* /Q command because it is in use. You will not lose any records for the current day.

My next task was to get the data into a database. Since my knowledge on databases is limited I decided the easiest way would be to try and import the data into a Microsoft Access database. I created a new Access database by selecting Blank Database when the program opened up. I named the database evtvwr. I then tried to import one of the dump files from Dumpev,t using the import wizard. I stepped through the wizard and when I selected the dump file to import, it recognized it as a comma delimited file and proceeded to separate the fields, name the fields, and import the data into the table I called Application. For the field names I chose: Log which identifies which log the event came from, Date for the date, Time for the time, Source for the source of the event, Eventid for the event identification number, Type for the type of event it represents, Category for the category the event belongs to, User for the user credentials that created the event, Computer for which server the event came from, and Message for the message generated for the event. This worked, but I immediately realized a problem. How was I going to automatically have the dump files imported into this Access database each day? I looked through the several Microsoft Access books, the Microsoft Access help files, and on the Microsoft web site but I could not find any tools that would allow me to do this operation. I did, however, know how to import data into a Microsoft SQL 2000 database using the BCP command

from my work writing batch file for the DBA Group. So I decided to use MSQL 2000 instead of Access. Looking back, I would have eventually had to make this same migration because of the shear number of entries and the size of the database.

MSQL 2000 is located on the central server KODOS. I opened up the Enterprise Manager and created a new database, called evtvwr, using the Database Creation Wizard. I made the database size 500mb and the log size 250mb, with the option to grow at a rate of 10%. I figured the easiest way to create the tables would be to import the Microsoft Access database evtvwr that I had already created. I selected from the Actions drop down menu, All Tasks and then Import Data. For the datasource, I chose Microsoft Access and selected the Access database I had created earlier. For the destination, I chose SQL Server as the datasource and the evtvwr as the database. I then selected the table I created in the Access database called Application and started the transfer. When the transfer completed, I checked the tables and saw the Application table listed within. I checked the properties and the following where the values assigned to the columns Name, Datatype, and Size:

Table1

[Log]	nvarchar	255
[Date]	datetime	8
[Time]	nvarchar	50
Source	nvarchar	255
Eventid	int	4
Type	nvarchar	255
Category	nvarchar	255
[User]	nvarchar	255
Computer	nvarchar	255
Message	ntext	16

I tested a dump file from Dumpevt to see if I could BCP it into the database. I looked at a batch file that I had previously written with the DBA group which imported data into another database. I changed the settings to reflex my database and the application table and the file I wanted to BCP in. The following is the syntax:

C:\progra~1\micro~1\80\tools\binn\bcp evtvwr.dbo.application in d:\eventlog\dumpevt\output*[servername]*sys.txt –U*[username]* –P*[password]* –KODOS –c –t","

For the username and password, I created a new login and assigned it db_owner privileges for the evtvwr database so it had sufficient rights to write to the database. The –c –t options allow the data to be imported into the database as character data with a comma separating the fields. I executed the command and received a successful return of 4596 rows copied. I tested to see if those events

actually did get into the database by clicking on the option to Return All Rows and checking to see if the entries for that server were in the logs. They were. I created a batch file called evtimport.bat to import all the dumped Eventviewer logs from each of the servers using the above syntax. I set up a scheduled task to run the batch file each night at 6:15am. This would allow enough time for the evtdump.bat to finish before the BCP commands started.

I needed to figure out a way to get the IIS and Urlscan logs into the database. During the testing for tools to import the Eventviewer logs, I came across a tool that could help me in this area. The Log Parser software actually converts IIS and Urlscan lots to a format that can be imported into a SQL database and performs the import automatically. I copied the Log Parser software to D:\Eventlog\Logparser to test it. Unfortunately, this tool is based on SQL statements which, I know very little about. However, Microsoft provides some very detailed documentation with this product which I downloaded from http://www.microsoft.com/windows2000/downloads/tools/logparser/default.asp. I figured I would start with the IIS logs. My logs are in the W3C Extended Log format. So I went to that section of the documentation. I recognized the fields in the table as some of the information I choose to log in my IIS log files. The table is below:

Table 2 [1]

Field	Data Type
LogFilename	STRING
LogRow	INTEGER
date	TIMESTAMP
time	TIMESTAMP
c-ip	STRING
cs-username	STRING
s-sitename	STRING
s-computername	STRING
s-ip	STRING
s-port	INTEGER
cs-method	STRING
cs-uri-stem	STRING
cs-uri-query	STRING
sc-status	INTEGER
sc-substatus	INTEGER
sc-win32-status	INTEGER
sc-bytes	INTEGER
cs-bytes	INTEGER

time-taken	INTEGER
cs-version	STRING
cs-host	STRING
cs(User-Agent)	STRING
cs(Cookie)	STRING
cs(Referer)	STRING
s-event	STRING
s-process-type	STRING
s-user-time	REAL
s-kernel-time	REAL
s-page-faults	INTEGER
s-total-procs	INTEGER
s-active-procs	INTEGER
s-stopped-procs	INTEGER

I looked back into my logs and found the fields that I actually use. They are Date, for the date the request was made, Time, for what time the request was made, C-ip, which is the IP address of the user accessing your website, Cs-username, which is the username if you require a login, S-ip, which is the IP address of the web site, S-port, which is the port that the web site is running on, Cs-method which, is the type of method used while interacting with the web site (i.e. GET, POST, etc.), Cs-uri-stem, which is the file name requested, Cs-uri-query, which is used by SMTP, Sc-status, which is the HTTP return code, and Cs(User-Agent), which is the type of browser used by the client. I then went to the following example and tried to piece it apart.
[2]

```
logparser "Select TO TIMESTAMP(date, time) as Timestamp,
cs-uri-stem as UriStem, cs-uri-query as UriQuery FROM ex000123.log TO TestTable" –
i:W3C -o:SQL -server:GABRIEGISQL
-driver:"SQL Server" -database:LogDB  -username:user
-password:xxx -createtable:ON
```

I figured the TO_TIMESTAMP parameter would import the date and time. The next part seemed like the fields listed in Table 2 with a new friendly name attached. The rest of it was self explanatory and the last field –createtable:ON seemed to me that this would actually create the table automatically. I went ahead and typed in the following command at the command prompt using one of my IIS logs:

[1,2] “Log Parser.” Microsoft Corporation. 2002. http://www.microsoft.com/windows2000/downloads/tools/logparser/default.asp

logparser "Select TO_TIMESTAMP(date, time) as Timestamp, c-ip as UserIP, cs-username as Username, s-ip as Sourceip, s-port as Sourceport, cs-method as Method, cs-uri-stem as Filerequest, cs-uri-query as Query, sc-status as Code, cs(User-Agent) as Browser FROM D:\eventlogs*[servername]*\w3svc1\ex030201.log TO iisevt" -i:W3C -o:SQL -server:kodos -driver:"SQL Server" -database:evtvwr -username:*[username]* -password:*[password]* –createtable:ON

I called the new table iisevt and I used the same user name and password I used in the evtimport.bat file. I executed the command and after a few seconds I got back a message saying, 3588 elements were processed. I verified that the data was imported into the database by going to tables and looking for iisevt. The fields of the table were as follows:

Table 3 [3]

[Timestamp]	datetime	8
Userip	varchar	256
Username	varchar	256
Sourceip	varchar	256
Sourceport	int	4
Method	varchar	256
Filerequest	varchar	256
Query	varchar	256
Code	int	4
Browser	varchar	256

I clicked on Return All Rows to verify the data. It returned the information from the log in the correct fields. I set up a batch file to logparse all the log files for IIS but I ran into a problem. How do you reference a fill name to import that is always changing? The IIS and Urlscan log file are named by the date they are created on. So I went back to the documentation and found that you can use a wildcard character (*) to select all logs. In the batch file called iis_logparser.bat, I entered in the following syntax for each server and each web instance on that server:

[3] "Log Parser." Microsoft Corporation. 2002. http://www.microsoft.com/windows2000/downloads/tools/logparser/default.asp

logparser "Select TO_TIMESTAMP(date, time) as Timestamp, c-ip as UserIP, cs-username as Username, s-ip as Sourceip, s-port as Sourceport, cs-method as Method, cs-uri-stem as Filerequest, cs-uri-query as Query, sc-status as Code, cs(User-Agent) as Browser FROM D:\eventlogs*[servername]*\w3svc*[instance #]**.log TO iisevt" -i:W3C -o:SQL -server:kodos -driver:"SQL Server" -database:evtvwr -username:*[username]* -password:*[password]*

I scheduled this to run as a scheduled task at 1:00am every morning. This would allow enough time for the IIS log files to be copied over from their server to KODOS. With this experience it was rather straight forward to set up the Urlscan logs in the same manner. The following is the table for the Urlscan fields:

Table 4

Field	Data Type
LogFileName	STRING
LogRow	INTEGER
Date	TIMESTAMP
ClientIP	STRING
Comment	STRING
SiteInstance	INTEGER
Url	STRING

Following the same logic, I ran the logparse command to import the data of an Urlscan log into a new table called Urlscan. The syntax is a follows

logparser "Select Date as Timestamp, ClientIP as UserIP, Comment as Description, SiteInstance as Instance, Url as Webaddress FROM D:\eventlogs*[servername]*\urlscan\urlscan.020103.log TO urlscan" -i:URLSCAN -o:SQL -server:kodos -driver:"SQL Server" -database:evtvwr -username:*[username]* -password:*[password]* –createtable:ON

When I executed the command, I got back a message saying 354 elements were processed. I verified that the data was imported into the database by going to tables and looking for Urlscan. It was there and the fields of the table were as follows:

Table 5

[Timestamp]	datetime	8
Userip	varchar	256
Description	varchar	256
Instance	int	4
Webaddress	varchar	256

I clicked on Return All Rows to verify the data. It returned the information from the log in the correct fields. I wrote a batch file called Urlscan_logparser.bat and entered the following command for each server.

```
logparser "Select Date as Timestamp, ClientIP as UserIP, Comment as
Description, SiteInstance as Instance, Url as Webaddress FROM
D:\eventlogs\[servername]\urlscan\*.log TO urlscan" -i:URLSCAN -o:SQL -
server:kodos -driver:"SQL Server" -database:evtvwr -
username:[username] -password:[password]
```

I scheduled this batch file as a scheduled task to run at 12:30am, giving the servers enough time to copy over their Urlscan logs to the central server.

With a complete database, it was time to import all the log files. I deleted the log files that were already imported into the database and then ran all three import scripts. Since these logs contained months worth of data it took several hours for all the logs to be imported. When it was finished, I had a database over 2 gig in size.

It was now time to do some cleanup. I wanted to have all the log files on the centralized server deleted so that the next time the import batch files were run they would not import duplicate records. But I also wanted to make sure that a backup copy of the IIS and Urlscan logs were taken from the nightly backup (we already have a backup of the Eventviewer logs). So I scheduled a batch file to run at 6:00am after the backups have been completed. I called the script nightly_delete.bat which deletes the IIS and Urlscan logs. The syntax is as follows:

```
cd D:\eventlog\logs\[servername]\w3svc1
del *.* /Q
cd D:\eventlog\logs\[servername]\w3svc2
del *.* /Q
[etc]
cd D:\eventlog\logs\[servername]\urlscan
del *.* /Q
```

I also created a batch file called deletedump.bat which deletes the Eventviewer dump files. This scheduled task runs at 6:45am, which gives the importation of the logs plenty of time to run. The syntax is as follows:

```
del D:\Eventlog\DumpEvt\output\*.* /Q
```

With that portion of the project complete, it was time to write queries to get the relevant data out of the. I first focused on the Eventviewer entries. The events that I wanted to look for on my servers were researched from an article called,

"Dealing with Windows NT Event Logs, Part One and Part Two," by Cory L Scott, 4,4,2000 located at, http://online.securityfocus.com/infocus/1334 and http://online.securityfocus.com/infocus/1335 and from the Security Operations Guide for Windows 2000 Server. They are as follows:

Table 6

Event 512	System Starting Up
Event 513	System Shutting Down
Event 516	Audit Event Records Disabled
Event 517	Audit Log Cleared
Event 529	Unknown user name or bad password
Event 531	Account Disabled
Event 532	Account Expired
Event 533	User not allowed to logon
Event 534	Logon type restricted
Event 535	Password Expired
Event 537	Unsuccessful logon
Event 539	The Account was locked out as the logon attempt was made
Event 577	A user attempted to perform a privileged system service operation
Event 608	User Right Assigned
Event 609	User Right Removed
Event 610	New Trusted Domian
Event 611	Remove Trusted Domain
Event 612	Audit Policy Changed
Event 614	IPSec Policy Agent Disabled
Event 615	IPSec Policy Changed
Event 616	IPSec Policy Agent encountered a potentially serious failure
Event 624	User Account Created
Event 627	Change Password Attempt
Event 628	User Account Password Reset
Event 629	User Account Disabled
Event 630	User Account Disabled
Event 636	Local Group Member Added
Event 644	User Account Locked Out
Event 6008	Shutdown was unexpected

Each of these events are important because they could indicate an attempted intrusion or successful penetration of your server. These events are not normal and every day activities in my environment. I wanted to write my queries to extract any records that had any of the above event ids in the Eventid field. Within Query Analyzer, I test the following query to see if I could extract the data I was looking for:

```
Select *
From application
Where eventid=529
```

This query worked and returned 19357 rows from the entire table. But what I really needed were the rows that occurred within the last day. This was because I did not want to receive the same information every day and have the output grow in size, where it will clog our email servers and become a burden to sort through. I only wanted the previous days events emailed to me so I could see what had happened on that day. I created the following query to see if I could get back information just for one day:

```
Select *
From application
Where eventide=529 and [date]='2003-01-31'
```

This query returned 106 rows from 2003-01-31. I needed to have the query scripted so that it would run automatically each day. I looked all over the Query Analyzer for a scheduling function but couldn't find one. I also checked the help file with but didn't find any mention of one. I decided to use a command line tool that I have seen the DBA's use, called osql.exe. I looked in the help file for the syntax for the command and found the following example:

```
osql /Q "EXIT(SELECT COUNT(*) FROM '%1')"
```

I replaced the part after the EXIT option with my query which became:

```
osql –U[username] –P[password] -Skodos -devtvwr /Q "EXIT(SELECT *
FROM application WHERE eventide=529 and [date]='2003-01-31')"
```

This worked great. Now I could automatically run the batch file as a scheduled task. But I ran into another problem. I needed the date to always reflect at least the current day or preferably the day before the current and not a static date. I decided to use the DOS date and insert it using the set command and a variable. I went to the regional settings on KODOS and changed the date format to represent yyyy-MM-dd, like it is in the database. I then created the following batch file:

```
Set date=%date%
osql –U[username] –P[password] -Skodos -devtvwr /Q "EXIT(SELECT *
FROM application WHERE eventide=529 and [date]='%date%')"
```

I ran the batch file and it came back with an error message stating:

```
Msg 241, Level 16, State 1, Server ERIE, Line 1
Syntax error converting datetime from character string.
```

I looked at the output on the command prompt and noticed that the date was in the correct format but that DOS also inserted the name of the Day before the date (i.e. Mon, Tue, etc). I searched the help files and Microsoft site but I could not find a way to prevent this from happening. So I needed to find another solution. After looking through the help file, I found a utility that looked promising called GETDATE. The examples did not show it being used in the Where statement but I figured I would give it a try. So, in Query Analyzer, I wrote the following query:

```
Select *
From application
Where eventid=529 and [date]=GETDATE()
```

When I executed the query I did not get any errors but I also did not get any rows returned. What I did get was the field names. I knew that there were Event 529 entries for the current date so I knew something was wrong. I went back to the help file for answers. I couldn't find any so I decided to run the simple example they had which was:

```
Select GETDATE()
Go
```

That was when I noticed that the GETDATE command not only returns the current date but the current time. This meant I could never get any events returned using the current date and time because I was running the query after the data had already been loaded. So I was back to square one. I again looked though the SQL book, through the help files and on the Microsoft web site but I couldn't find anything. I then decided to ask the DBA group for suggestions. Unfortunately, they could not come up with any. Upon further research, I came across an interesting command called DATEADD in the help file. The Microsoft SQL 2000 Help file states that the tool returns a new datetime value based on adding an interval to the specified date. I reasoned out that I might be able to input the GETDATE utility for the Date argument and subtract a day from it by using a negative one for the Number argument. Then I would be able to return all records that were greater than that date. Here is the DATEADD syntax:

```
DATEADD (datepart,number,date)
```

I went back to the query analyzer and typed in the following syntax:

```
SELECT *
FROM application
WHERE (eventid=529) and ([date]>dateadd(day,-1,getdate()))
```

Notice the ">" sign, which allows all the events with dates greater than the current date and time, minus 24 hours. I ran this query and I received 102 rows back.

Now I had the syntax down to automatically run the query. I wanted the results of the query to be in a file so it could be emailed to me. So I created the following batch file to test it:

> osql –U*[username] –P[password]* -Skodos -devtvwr /Q "EXIT(SELECT * FROM application WHERE eventide=529 and ([date]>dateadd(day,-1,getdate())))" > text.txt

The command should send the results of the osql statement to the file text.txt. I executed the command and watched the text.txt file being created. To my surprise, it grew and grew and grew. It ended up being over 14mb in size and that was for only 102 records. When it finished, I opened up the file and saw why the file was so big. The formatting included an incredible amount of blank space between most of the fields. I figured it had to be because of the size of the fields. I went back to the application tables and clicked on the Design Table option so I could fix the size and produce an output file that was easily readable.

As shown in Table 1, all the nvarchar fields were 255 in size. I thought all these fields were too big so I went back to the actual dump files and opened them up looking for the largest entries in each field. After looking at numerous logs, I figured I could reduce the size of most of the fields. But since I had done all this work on the database I was afraid to change any of its structure, in fear of losing the data or corrupting the whole database. So I ask the DBA group to look at this for me. They told me that I could change the field sizes but questioned why the data types were nvarchar and ntext. I told them that it was automatically generated that way. They suggested that I change all the fields, except the date field, to varchar which is a character based data type that doesn't pad the field with spaces. Since I wasn't doing any calculations on the data this seemed like a great idea and would solve the blank space problem in my query outputs. The date field would remain a datetime data type. I went in and changed all the fields. The following table shows the results.

Table 7

[Log]	varchar	3
[Date]	datetime	8
[Time]	varchar	8
Source	varchar	40
Eventid	varchar	5
Type	varchar	11
Category	varchar	25
[User]	varchar	50
Computer	varchar	13
Message	varchar	1000

After the database was updated, I ran the query again and directed it to the test.txt file. The size of the file reduced from 14mb to 867kb. When I opened it up it was perfectly formatted so that each event was on a single line. I decided to do the same thing to the iisevt and Urlscan database tables. I changed the iisevt from Table 3 to the follow configuration:

Table 8

[Timestamp]	datetime	8
Userip	varchar	15
Username	varchar	25
Sourceip	varchar	15
Sourceport	int	4
Method	varchar	10
Filerequest	varchar	100
Query	varchar	125
Code	int	4
Browsre	varchar	100

I changed the Urlscan database table from Table 5 to the following configuration:

Table 9

[Timestamp]	datetime	8
Userip	varchar	15
Description	varchar	100
Instance	int	4
Webaddress	varchar	120

I had to leave the data types the same because Log Parser requires specific data types for those fields as described in Table 2 and Table 4.

With all those changes I noticed that my database and log file had doubled in size without putting any more information into them. I had seen a Shrink Database option earlier on one of the pull down tabs so I looked it up in the help file. It seemed like it would do the trick by getting rid the free space. I ran the command and selected the defaults. The database shrank from 5 gigs to 1.5 gigs in size. I thought that this would probably need to be done routinely so I decided to set up a maintenance plan. I found Database Maintenance under Management in the Enterprise Manager and selected a New Maintenance Plan. The following are the options I chose to do daily:

Reorganize data and index pages
    Change free space per page to 10%
Remove unused space from database files
    Shrink database when it grows beyond 3000mb
    Amount of free space to remain after shrink – 10%

Check database integrity
Include indexes
Backup database as part of maintenance plan
Verify integrity of backup upon completion
To Disk
Use default backup directory
Remove files older than 4 weeks
Backup transaction logs as part of maintenance plan
Verify integrity of backup upon completion
To Disk
Use default backup directory
Remove files older than 4 weeks
Write report to a text file in directory D:\eventlog\results\

While I was looking for the maintenance plan, I stumbled across the SQL Server Agent in the Enterprise Manager, under Management. This had a utility called Jobs in it. Here was the answer to my scheduled queries all a long. I no longer needed to use the osql.exe utility. This also solved another dilemma that I was facing. Each of the SQL batch files I wrote contained the user name and password to the database in clear text! I could now write all those commands within the Jobs function and not worry about a specific file being compromised.

I created a new Job call Eventviewer Nightly Returns and selected the owner as the login account I created earlier. Under steps, I decided to create a single step for each of the events listed in Table 6. This way, I could get a single output file for each event which would make it easier to identify what I was looking at when the results were mailed to me. For each "step" I gave it the name of the Event I was querying upon. For the Type, I chose Transact-SQL Script (TSQL) since I was doing SQL scripts. I selected the database evtvwr and used the following command:

```
SELECT *
FROM application
WHERE (eventid=529) and ([date]>dateadd(day,-1,getdate()))
```

I clicked on the Advanced Tab and selected to output the file to D:\eventlog\Results\ where I would keep the results for everything I scheduled. I named the file event529.txt. I then saved the Job and ran it. I looked in the Results directory and found the file, properly formatted with events listed in it. I went back to the Job and entered in a "step" for each of the events from Table 6 with its own output file. I then scheduled the Job to run at 7:00am every morning leaving enough time for all the dump files to get imported into the database.

Now it was time to write the queries for IIS. I wanted to know who was trying to access the web sites but received error messages in return from the web server. I decided to key on the HTTP error codes. I went to the location of the error code

.htm files on one of the web servers at C:\WINNT\Help\iisHelp\common and came up with the following numbers to query on.

HTTP 403
HTTP 400
HTTP 401.1
HTTP 401.2
HTTP 401.3
HTTP 401.4
HTTP 401.5
HTTP 403.1
HTTP 403.2
HTTP 403.3
HTTP 403.4
HTTP 403.5
HTTP 403.6
HTTP 403.7
HTTP 403.8
HTTP 403.9
HTTP 403.10
HTTP 403.11
HTTP 403.12
HTTP 403.13
HTTP 403.14
HTTP 403.15
HTTP 403.16
HTTP 403.17
HTTP 404.1
HTTP 404
HTTP 405
HTTP 406
HTTP 407
HTTP 410
HTTP 412
HTTP 414
HTTP 500
HTTP 501
HTTP 502

When looking at the logs, I noticed that the IIS logs contain information that is also reported in the Urlscan logs. I did not want to get duplicate information so I needed to figure out a way to make sure anything rejected by Urlscan was not reported in the query results. I did this using NOT argument in the query statement. Anytime Urlscan rejected a request it would report /<Rejected-By-UrlScan> in the Filerequest column of the iisevt table. I also wanted only the previous day's results. I developed the following query:

```
select *
from iisevt
where ([timestamp]>dateadd(day,-1,getdate())) and not
(filerequest='/<Rejected-By-UrlScan>') and ((code=403) or (code=400) or
(code=401.1) or (code=401.2) or (code=401.3) or (code=401.4) or
(code=401.5) or (code=403.1) or (code=403.2) or (code=403.3) or
(code=403.4) or (code=403.5) or (code=403.6) or (code=403.7) or
(code=403.8) or (code=403.9) or (code=403.10) or (code=403.11) or
(code=403.12) or (code=403.13) or (code=403.14) or (code=403.15) or
(code=403.16) or (code=403.17) or (code=404.1) or (code=404) or
(code=405) or (code=406) or (code=407) or (code=410) or (code=412) or
(code=414) or (code=500) or (code=501) or (code=502))
```

When I ran the query it gave me all the entries that had those error codes and the events were all from the previous day. I created a Job called IIS Nightly Returns and selected the same parameters as I did in the Eventviewer Job. I entered in the above query at the command option and I selected to output the results to D:\eventlog\Results\iislogs.txt. I wanted these returns all in one file so it would be easier to spot patterns on the denied requests. I scheduled the job to run at 2:00am, giving enough time for the IIS logs to be imported since their size can be quite large.

For the Urlscan logs, I wanted to get a report each day on everything Urlscan rejected from the previous days activities. So I created a Job called Urlscan Returns and selected as my query the following command:

```
select *
from urlscan
where ([timestamp]>dateadd(day,-1,getdate()))
```

This returned all events from the previous day. I had the results sent to D:\eventlog\Results\urlscan.txt. I scheduled this job to run at 12:55am giving enough time for the Urlscan logs to be imported.

I went back to the three scripts that imported the data into the database with the clear username and password (evtimport.bat, iis_logparser.bat, and Urlscan_logparser.bat) and created an individual job for each in Enterprise Manager with the same exact syntax and same scheduled time as the batch file. The only difference between these three new Jobs and the previous SQL queries was that I selected for the Type of operation Operating System Command (CmdExec), instead of Transact-SQL Script (TSQL), since these are command line tools and not SQL queries. I set these three jobs up to send an output file to D:\eventlog\results\ called dumpevtresults.txt, iisresult.txt and urlscanresults.txt respectfully.

I then ran each of the jobs manually. I looked in the results directory and saw 35 text files. I opened each of them to make sure the content was correct. Everything looked great. Now it was time to have the files emailed to me.

I have used a product called BLAT to email me text documents before so I decide to use it to email all the files located in the results directory. I copied the utility over to the directory D:\eventlog\blat. I then created a batch file called resultsmail.bat. I entered in the following command:

> D:\eventlog\blat\blat D:\eventlog\blat\evtlogfiles.txt -t *[name@domainname.com]* -s "Event Logs" -mime -attach D:\eventlog\results*.txt

The eventlogfiles.txt file is needed to fill the message field. This file must exist. I created that file with the following text, “Daily Event Logs for Review,” and saved the file. The –t option, is who the mail should be sent to. The –s option, is the subject line. The –mime option, is the type of email. And the –attach, option specifies what files to attach. I ran this batch file and a minute later I received the email with all the result files attached. I then scheduled the batch file to run at 7:30am after all the Jobs have run and before I arrive in the morning.

After

The centralized event log server not only brings the server farm up to compliance with our log policy but it also makes our organization more secure. The time and effort it used to take to filter out the events that concerned me most can now be spent elsewhere, securing and monitoring other aspects of my systems. I now get daily reports of all the important activities on my servers in the morning that may have been over looked in the past. I can easily see which servers are getting the most hacker attention and what methods are being used. And I also have an easy and reliable way to backup and archive all our event logs in a secure location preventing a hacker from altering past activities.

With the centralized log server I have already discovered a scheduled task that was not executing because of a bad password that was backing up a servers system state data each night. I have seen repeated attempts by multiple IP addresses using Code Red and Nimda, which I have blocked locally using IPSec and sent to the network people to block at the firewall. If the IP addresses are local, I can contact those people who own the system and inform them that they have a virus. I have also seen many spider entries in the IIS logs looking for the robots.txt file. These have also been blocked using IPSec.

I will take these results and combine them with real time tools like NTLast and EventcombMT, that I have tested previously. I can track events happening right now, like failed login attempts, IPSec policy changes, a privileged system

operation performed, etc. This will allow me to have nearly complete knowledge of what is happening on my servers.

References

Software

Dumpevt http://www.systemtools.com/somarsoft/

Dumpel http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/dumpel-o.asp

NTLast http://www.foundstone.com/knowledge/proddesc/ntlast.html

NTOLog http://www.ntobjectives.com/products/firewater/

Eventsave http://www.heysoft.de/Frames/f_sw_es_en.htm

Eldump http://www.ibt.ku.dk/jesper/eldump/eldump.htm

Elsavclr http://www.ibt.ku.dk/jesper/elsavClr/

Log Parser http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=8CDE4028-E247-45BE-BAB9-AC851FC166A4

Log Agent http://www.geocities.com/floydian_99/

EventcombMT http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=9989D151-5C55-4BD3-A9D2-B95A15C73E92

*Book*s

Praque, Cary N. and Irwin, Michael R. with Reardon, Jennifer. Microsoft Access 2000 Bible. IDG Books Worldwide, Inc, 1999.

Jennings, Rodger. Special Edition Using Microsoft Access 2000.Que Corporation, 1999.

Vieira, Robert. Professional SQL Server 2000 Programming. Wrox Press, 2000

Articles

"Log Parser." Microsoft Corporation. 2002.
http://www.microsoft.com/windows2000/downloads/tools/logparser/default.asp

"Microsoft Log Parser 2." IISFAQ. November 25, 2002.
http://www.iisfaq.com/default.aspx?View=A525&P=1

Smith, Randy. "Archiving and Analyzing the NT Security Log." Windows & Net Magazine. http://www.win2000mag.com/Articles/Index.cfm?ArticleID=9043

Floydman. "Log Agent, Log File Recollection Tool."
http://iquebec.ifrance.com/securit/whitepapers/logagent.html

"Security Operations Guide for Windows 2000 Server." Microsoft Corporation.
Ihttp://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/windows/windows2000/staysecure/default.asp

Scott, Cory L. "Dealing with Windows NT Event Logs, Part One." April 4, 2000.
http://online.securityfocus.com/infocus/1334
Scott, Cory L. "Dealing with Windows NT Event Logs, Part Two." April 4, 2000.
http://online.securityfocus.com/infocus/1335

"Microsoft Access Help." Microsoft Corporation. 1992 – 1999.

"Microsoft SQL Server Books Online." Microsoft Corporation. 1988 - 2000.

Websites

www.google.com

www.microsoft.com