Monitoring Microsoft SQL Server Audit Logs with EventTracker

The Importance of Consolidation, Correlation, and Detection

Enterprise Security Series
Abstract

The purpose of this paper is to highlight the major advantages of employing EventTracker to consolidate and manage Microsoft SQL Server Enterprise Edition audit log data. The paper introduces at a high level, the major design concepts that enable EventTracker to process, store and allow users to gain actionable intelligence from the millions of critical events generated by SQL Server.

SQL Server event data contains a wealth of valuable information for both Database Administrators and for security controls and compliance. Monitoring and managing SQL Server audit logs manually is tedious, time consuming and for all practical purposes impossible for a large setup. This paper explains how to set up auditing so that EventTracker can effectively and efficiently perform these jobs.

The steps mentioned here for setting up the SQL Server (Enterprise Edition only) audit trail are for the Windows Operating System.

Scope

# Table of Contents

Abstract.................................................................................................................................................. 1
Scope .................................................................................................................................................. 1
SQL Server Audit Setup.......................................................................................................................... 3
Components of Auditing .......................................................................................................................... 3
  SQL Server Audit ................................................................................................................................. 3
  Server Level Audit Specification ....................................................................................................... 3
  Database Level Audit Specification ................................................................................................. 3
  Target .............................................................................................................................................. 3
Configure SQL server audit ................................................................................................................ 4
Audit Action Categories....................................................................................................................... 9
  Server Level ..................................................................................................................................... 9
  Database Level ............................................................................................................................... 11
  Audit Level .................................................................................................................................... 12
Configure EventTracker Filter Exception ............................................................................................. 12
SQL Audit Log Format.......................................................................................................................... 15
Monitoring, Alerting and Reporting with EventTracker ......................................................................... 17
  Monitoring ........................................................................................................................................ 17
  Alerting .......................................................................................................................................... 18
  Reporting ...................................................................................................................................... 19
The EventTracker Solution................................................................................................................... 22
About Prism Microsystems.................................................................................................................... 24
SQL Server Audit Setup

Tracking and logging database activity is critical for overall Enterprise security and in addition provides data for many compliance requirements such as Sarbanes Oxley, SAS 70, etc. In SQL Server 2008, 2012 and 2014, Microsoft has introduced new auditing capability that is invaluable to the DBA tasked with producing detailed audits that track database usage.

Through this new audit capability auditing can be implemented at the server and database levels, enabled on individual database objects and saved to different formats such as binary files or to the Windows Application or Security log. For EventTracker to collect, manage and monitor the SQL Server logs, it is necessary to forward the SQL Server logs to either the Windows Application or Security logs.

This White Paper describes the steps necessary to setup the auditing functions.

Components of Auditing

There are 4 components to the audit system:

- **SQL Server Audit** - The SQL Server Audit object is the audit component that collects Server or Database level actions or group of actions.

- **Server Level Audit Specification** - The Server Level Audit Specification object is the audit component that collects server level actions or group of actions.

- **Database Level Audit Specification** - The Database Level Audit Specification object is the audit component that collects database level actions or group of actions.

- **Target** - The Target component is the destination where the audit records are written. It could either be the Windows Application or Security log or a binary file.
Configure SQL server audit

- Go to Object Explorer in MS SQL Server Management Studio.

- In the object explorer, navigate to Security>Audits.

- Right click on Audit and select New Audit.

This opens Create New Audit page.
In the **Audit Name** field, specify the name for the Audit Object.

In the **Audit destination** field, select **Application Log**.

Click **OK** to save.

In the object explorer, navigate to **Security>Server Audit Specification**.
• Right click on **Server Audit Specification** and click on **New Server Audit Specification**.

This opens **Create New Server Audit Specification** window.
In the Name field, specify the name for the Server audit.
Select your Audit object from the Audit field drop down menu.
In the Actions table, select the following Audit Action type from the list.

a) FAILED_LOGIN_GROUP
b) SUCCESSFUL_LOGIN_GROUP
c) DATABASE_OBJECT_CHANGE_GROUP
d) DATABASE_PRINCIPAL_CHANGE_GROUP
e) SCHEMA_OBJECT_CHANGE_GROUP
f) SERVER_PRINCIPAL_CHANGE_GROUP
g) LOGIN_CHANGE_PASSWORD_GROUP
h) SERVER_STATE_CHANGE_GROUP
i) DATABASE_ROLE_MEMBER_CHANGE_GROUP
j) SERVER_ROLE_MEMBER_CHANGE_GROUP
k) DATABASE_PERMISSION_CHANGE_GROUP
l) SERVER_OBJECT_PERMISSION_CHANGE_GROUP

- Click on **OK** to save.

![Object Explorer](image)

**Figure 5**

- Click on **Audits** node in **Object Explorer** and right click on the audit object created, and then select **Enable Audit**.
Click on **Server Audit Specification** node in **Object Explorer** and right click on the audit object created, and then select **Enable Server Audit Specification**.

EventTracker will now collect MS SQL audit events from the Windows Application log.

**NOTE** - Select actions in Audit specification as per your requirement. Actions available for audit specification have been enlisted below:

### Audit Action Categories

Audit actions can be grouped into following categories:

- **Server Level** – It includes server operations.

  Server Level Audit actions include the following:
1. SUCCESSFUL_LOGIN_GROUP
2. LOGOUT_GROUP
3. FAILED_LOGIN_GROUP
4. LOGIN_CHANGE_PASSWORD_GROUP
5. APPLICATION_ROLE_CHANGE_PASSWORD_GROUP
6. SERVER_ROLE_MEMBER_CHANGE_GROUP
7. DATABASE_ROLE_MEMBER_CHANGE_GROUP
8. BACKUP_RESTORE_GROUP
9. DBCC_GROUP
10. SERVER_OPERATION_GROUP
11. DATABASE_OPERATION_GROUP
12. AUDIT_CHANGE_GROUP
13. SERVER_STATE_CHANGE_GROUP
14. SERVER_OBJECT_CHANGE_GROUP
15. SERVER_PRINCIPAL_CHANGE_GROUP
16. DATABASECHANGE_GROUP
17. DATABASE_OBJECT_CHANGE_GROUP
18. DATABASE_PRINCIPAL_CHANGE_GROUP
19. SCHEMA_OBJECT_CHANGE_GROUP
20. SERVER_PRINCIPAL_IMPERSONATION_GROUP
21. DATABASE_PRINCIPAL_IMPERSONATION_GROUP
22. SERVER_PRINCIPAL_IMPERSONATION_GROUP
23. DATABASE_PRINCIPAL_IMPERSONATION_GROUP
24. SERVER_OBJECT_OWNERSHIP_CHANGE_GROUP
25. DATABASE_OWNERSHIP_CHANGE_GROUP
26. DATABASE_OBJECT_OWNERSHIP_CHANGE_GROUP
27. SCHEMA_OBJECT_OWNERSHIP_CHANGE_GROUP
28. SERVER_PERMISSION_CHANGE_GROUP
29. SERVER_OBJECT_PERMISSION_CHANGE_GROUP
30. DATABASE_PERMISSION_CHANGE_GROUP
31. DATABASE_OBJECT_PERMISSION_CHANGE_GROUP
32. SCHEMA_OBJECT_PERMISSION_CHANGE_GROUP
33. DATABASE_OBJECT_ACCESS_GROUP
34. SCHEMA_OBJECT_ACCESS_GROUP
35. BROKER_LOGIN_GROUP
36. DATABASE_MIRRORING_LOGIN_GROUP
37. TRACE_CHANGE_GROUP

- **Database Level** - Includes DDL and DML.

Database Level Audit actions include the following:

1. DATABASE_ROLE_MEMBER_CHANGE_GROUP
2. DATABASE_OPERATION_GROUP
3. DATABASE_CHANGE_GROUP
4. DATABASE_OBJECT_CHANGE_GROUP
5. DATABASE_PRINCIPAL_CHANGE_GROUP
6. SCHEMA_OBJECT_CHANGE_GROUP
7. DATABASE_PRINCIPAL_IMPERSONATION_GROUP
8. DATABASE_OWNERSHIP_CHANGE_GROUP
9. DATABASE_OBJECT_OWNERSHIP_CHANGE_GROUP
10. SCHEMA_OBJECT_OWNERSHIP_CHANGE_GROUP
11. DATABASE_PERMISSION_CHANGE_GROUP
12. DATABASE_OBJECT_PERMISSION_CHANGE_GROUP
13. SCHEMA_OBJECT_PERMISSION_CHANGE_GROUP
14. DATABASE_OBJECT_ACCESS_GROUP
15. SCHEMA_OBJECT_ACCESS_GROUP

- **Audit Level** – It includes object audit.

Audit Level Audit actions include the following:

1. AUDIT_CHANGE_GROUP

**Configure EventTracker Filter Exception**

![EventTracker Control Panel](image_url)

*Figure 7*
- Go to **EventTracker Agent Configuration** in EventTracker Control Panel.

![EventTracker Agent Configuration](image)

- Navigate to **Event Filters>Filter Exception**.

Filter Exception window opens.

Figure 9

- Click **New** to add exception.

Edit Event Details window opens.

Figure 10
• Configure options as suggested below:

1. Log Type – Application
2. Event Type – Information
3. Event ID – 33205
4. Match in Source – MSSQLSERVER

• Click OK to save.

SQL Audit Log Format

The target audit file consists of zero or more audit action items which are recorded. As mentioned previously, the target component is the destination where the audit records are written. The following are the elements in the records sent to the target.

• event_time
• sequence_no
• action_id
• succeeded
• permission_bitmask
• is_column_permission
• session_id
• server_principal_id
• database_principal_id
• object_id
• target_server_principal_id
• target_database_principal_id
• class_type
• session_server_principal_name

- server_principal_name
- server_principal_sid
- database_principal_name
- target_server_principal_name
- target_server_principal_sid
- target_database_principal_name
- server_instance_name
- database_name
- schema_name
- object_name
- TSQL statement
- additional_information

Below is an example of an Audit Event from SQL Server in EventTracker:
Once MSSQL Server is configured to send audit logs to either the Windows Application log or Windows Security log, the EventTracker agent resident on the system will transmit these logs to the EventTracker Console where they will be processed and stored. EventTracker provides support for efficiently monitoring, alerting and reporting of MSSQL Server Audit Logs.

Monitoring

EventTracker monitors all events generated by SQL Server. When DBA performs Log search with Source as MSSQLSERVER, SQL server events collected by EventTracker as shown as follows:
Alerting

EventTracker can alert DBA on critical events such as table deletion events, table alteration, user creation, user deletion, user modification and user role change. These alerts can be received via email, pager, SNMP traps, and as RSS feeds.
Reporting

EventTracker provides an exclusive reporting tool to generate requirement specific reports. Manual logging makes it difficult to retrieve the list of table alteration and user modification events. Below are sample reports created by EventTracker specific to SQL Server logs.

### MSSQLServer-User created

<table>
<thead>
<tr>
<th>LogTime</th>
<th>Instance Name</th>
<th>Database Name</th>
<th>Created By</th>
<th>User Name</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/23/2015 02:56:03 PM</td>
<td>ESXYMIN2K12R2VM2</td>
<td>master</td>
<td>TOONS\molly</td>
<td>testLogin</td>
<td>CREATE LOGIN [testLogin] WITH PASSWORD = *********, DEFAULT_DATABASE = [master], CHECK_EXPIRATION = OFF, CHECK_POLICY = OFF</td>
</tr>
<tr>
<td>12/23/2015 02:58:07 PM</td>
<td>ESXYMIN2K12R2VM2</td>
<td>master</td>
<td>TOONS\molly</td>
<td>testLogin</td>
<td>CREATE LOGIN [testLogin] WITH PASSWORD = *********, DEFAULT_DATABASE = [master], CHECK_EXPIRATION = OFF, CHECK_POLICY = OFF</td>
</tr>
<tr>
<td>12/23/2015 04:22:59 PM</td>
<td>ESXYMIN2K12R2VM2</td>
<td>master</td>
<td>TOONS\molly</td>
<td>testLogin</td>
<td>CREATE LOGIN [testLogin] WITH PASSWORD = *********, DEFAULT_DATABASE = [master], CHECK_EXPIRATION = OFF, CHECK_POLICY = OFF</td>
</tr>
<tr>
<td>12/23/2015 04:25:00 PM</td>
<td>ESXYMIN2K12R2VM2</td>
<td>master</td>
<td>TOONS\molly</td>
<td>testLogin</td>
<td>CREATE LOGIN [testLogin] WITH PASSWORD = *********, DEFAULT_DATABASE = [master], CHECK_EXPIRATION = OFF, CHECK_POLICY = OFF</td>
</tr>
<tr>
<td>12/23/2015 04:31:03 PM</td>
<td>ESXYMIN2K12R2VM2</td>
<td>master</td>
<td>TOONS\molly</td>
<td>LarryKing</td>
<td>CREATE LOGIN [LarryKing] WITH PASSWORD = *********, MUST_CHANGE, DEFAULT_DATABASE = [master], CHECK_EXPIRATION = ON, CHECK_POLICY = ON</td>
</tr>
<tr>
<td>12/23/2015 04:53:20 PM</td>
<td>ESXYMIN2K12R2VM2</td>
<td>master</td>
<td>TOONS\molly</td>
<td>testLogin</td>
<td>CREATE LOGIN [TOONS\molly] WITH PASSWORD = *********, MUST_CHANGE, DEFAULT_DATABASE = [master], CHECK_EXPIRATION = ON, CHECK_POLICY = ON</td>
</tr>
</tbody>
</table>

Figure 14

### MSSQLServer-User deleted

<table>
<thead>
<tr>
<th>LogTime</th>
<th>Database Name</th>
<th>Instance Name</th>
<th>Deleted By</th>
<th>User Name</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/23/2015 04:24:02 PM</td>
<td>master</td>
<td>ESXYMIN2K12R2VM2</td>
<td>TOONS\sally</td>
<td>testLogin</td>
<td>DROP LOGIN [testLogin]</td>
</tr>
<tr>
<td>12/23/2015 04:25:50 PM</td>
<td>master</td>
<td>ESXYMIN2K12R2VM2</td>
<td>TOONS\sally</td>
<td>testLogin</td>
<td>DROP LOGIN [testLogin]</td>
</tr>
<tr>
<td>12/23/2015 04:52:12 PM</td>
<td>master</td>
<td>ESXYMIN2K12R2VM2</td>
<td>TOONS\sally</td>
<td>LarryKing</td>
<td>DROP LOGIN [LarryKing]</td>
</tr>
<tr>
<td>12/23/2015 07:38:14 PM</td>
<td>master</td>
<td>ESXYMIN2K12R2VM2</td>
<td>TOONS\sally</td>
<td>TOONS\molly</td>
<td>DROP LOGIN [TOONS\molly]</td>
</tr>
<tr>
<td>12/23/2015 12:51:30 PM</td>
<td>master</td>
<td>ESXYMIN2K12R2VM2</td>
<td>TOONS\sally</td>
<td>testLogin</td>
<td>DROP LOGIN test</td>
</tr>
<tr>
<td>12/23/2015 02:38:44 PM</td>
<td>master</td>
<td>ESXYMIN2K12R2VM2</td>
<td>TOONS\sally</td>
<td>NicksSin</td>
<td>DROP USER [NicksSin]</td>
</tr>
<tr>
<td>12/23/2015 02:38:51 PM</td>
<td>master</td>
<td>ESXYMIN2K12R2VM2</td>
<td>TOONS\sally</td>
<td>checkyourmath</td>
<td>DROP LOGIN [checkyourmath]</td>
</tr>
<tr>
<td>12/23/2015 02:40:51 PM</td>
<td>master</td>
<td>ESXYMIN2K12R2VM2</td>
<td>TOONS\sally</td>
<td>Niks</td>
<td>DROP LOGIN Niks</td>
</tr>
<tr>
<td>12/23/2015 02:47:19 PM</td>
<td>master</td>
<td>ESXYMIN2K12R2VM2</td>
<td>TOONS\sally</td>
<td>NicksSin</td>
<td>DROP USER NicksSin</td>
</tr>
</tbody>
</table>

Figure 15
### MSSQLServer-User modified

<table>
<thead>
<tr>
<th>Log Time</th>
<th>Database Name</th>
<th>Instance Name</th>
<th>Changed By</th>
<th>User Name</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/29/2015 07:40:33 PM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>NkosSin</td>
<td>ALTER LOGIN [NkosSin] DISABLE</td>
</tr>
<tr>
<td>12/29/2015 10:18:04 AM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>Larry</td>
<td>ALTER LOGIN [Larry] WITH DEFAULT_DATABASE=[msdb], DEFAULT_LANGUAGE=[en], CHECK_EXPIRATION=ON, CHECK_POLICY=ON</td>
</tr>
<tr>
<td>12/29/2015 10:18:04 AM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>Moby</td>
<td>ALTER LOGIN [Moby] WITH DEFAULT_DATABASE=[msdb], DEFAULT_LANGUAGE=[en], CHECK_EXPIRATION=ON, CHECK_POLICY=ON</td>
</tr>
<tr>
<td>12/29/2015 10:16:04 AM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>NkosSin</td>
<td>ALTER LOGIN [NkosSin] WITH DEFAULT_DATABASE=[msdb], DEFAULT_LANGUAGE=[en], CHECK_EXPIRATION=ON, CHECK_POLICY=ON</td>
</tr>
<tr>
<td>12/29/2015 10:16:04 AM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>Sid</td>
<td>ALTER LOGIN [Sid] WITH DEFAULT_DATABASE=[msdb], DEFAULT_LANGUAGE=[en], CHECK_EXPIRATION=ON, CHECK_POLICY=ON</td>
</tr>
<tr>
<td>12/29/2015 10:54:07 AM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>NkosSin</td>
<td>ALTER LOGIN NkosSin</td>
</tr>
</tbody>
</table>

**Figure 16**

### MSSQLServer-User role changed

<table>
<thead>
<tr>
<th>Log Time</th>
<th>Database Name</th>
<th>Instance Name</th>
<th>Changed By</th>
<th>User Name</th>
<th>Role Name</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/29/2015 02:58:07 PM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>testLogin</td>
<td>sysadmin</td>
<td>alter server role [sysadmin] add member [testLogin]</td>
</tr>
<tr>
<td>12/29/2015 02:22:59 PM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>testLogin</td>
<td>sysadmin</td>
<td>alter server role [sysadmin] add member [testLogin]</td>
</tr>
<tr>
<td>12/29/2015 04:24:47 PM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>testLogin</td>
<td>discreator</td>
<td>ALTER SERVER ROLE [discreator] ADD MEMBER [testLogin]</td>
</tr>
<tr>
<td>12/29/2015 04:25:08 PM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>testLogin</td>
<td>db_securityadmin</td>
<td>ALTER ROLE [db_securityadmin] ADD MEMBER [testLogin]</td>
</tr>
<tr>
<td>12/29/2015 04:25:08 PM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>testLogin</td>
<td>db_owner</td>
<td>ALTER ROLE [db_owner] ADD MEMBER [testLogin]</td>
</tr>
<tr>
<td>12/29/2015 04:31:03 PM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>LarryKing</td>
<td>sysadmin</td>
<td>ALTER SERVER ROLE [sysadmin] ADD MEMBER [LarryKing]</td>
</tr>
<tr>
<td>12/29/2015 07:35:00 PM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>TOONSswalter</td>
<td>serveradmin</td>
<td>ALTER SERVER ROLE [serveradmin] DROP MEMBER [TOONSswalter]</td>
</tr>
<tr>
<td>12/29/2015 07:35:00 PM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>TOONSswalter</td>
<td>securityadmin</td>
<td>ALTER SERVER ROLE [securityadmin] DROP MEMBER [TOONSswalter]</td>
</tr>
<tr>
<td>12/29/2015 07:35:00 PM</td>
<td>master</td>
<td>ESXWIN2K12R2VM2</td>
<td>TOONSswalter</td>
<td>TOONSswalter</td>
<td>db_datawriter</td>
<td>ALTER ROLE [db_datawriter] ADD MEMBER [TOONSswalter]</td>
</tr>
</tbody>
</table>

**Figure 17**
The EventTracker Solution

The EventTracker solution is a scalable, enterprise-class Security Information and Event Management (SIEM) solution for Windows systems, Syslog/Syslog NG (UNIX and many networking devices), SNMP V1/2, legacy systems, applications and databases. EventTracker enables "defense in depth", where log data is automatically collected, correlated and analyzed from the perimeter security devices down to the applications and databases. To prevent security breaches, event log data becomes most useful when interpreted in near real time and in context. Context is vitally important because often the critical indications of impending problems and security violations can only be learned by watching patterns of events across multiple systems. Complex rules can be run on the event stream to detect signs of such a breach. EventTracker also provides real-time alerting capability in the form of an email, page or SNMP message to proactively alert security personnel to an impending security breach.

The original event log data is also securely stored in a highly compressed event repository for compliance purposes and later, forensic analysis. For compliance, EventTracker provides a powerful reporting interface, scheduled or on-demand report generation, automated compliance workflows that prove to auditors that reports are being reviewed and many other features. With pre-built auditor grade reports included for most of the compliance standards (FISMA, HIPAA, SOX, GLBA, and NISPOM); EventTracker represents a compliance solution that is second to none. EventTracker also provides advanced forensic capability where all the stored logs can be quickly searched through a powerful Google-like search interface to perform quick problem determination.

EventTracker lets users completely meet the logging requirements specified in NIST SP 800-92 Guide To Computer Security Log Management, and additionally provides Host Based Intrusion Detection, Change Monitoring and USB activity tracking on Windows systems, all in a turnkey, off the shelf, affordable, software solution.

EventTracker provides the following benefits

- A highly scalable, component-based architecture that consolidates all Windows, SNMP V1/V2, legacy platforms, Syslog received from routers, switches, firewalls, critical UNIX servers (RedHat Linux, Solaris, AIX etc), Solaris BSM, workstations and various other SYSLOG generating devices.

- Automated archival mechanism that stores activities over an extended period to meet auditing requirements. The complete log is stored in a highly compressed (>90%), secured archive that is limited only by the amount of disk storage.
• Real-time monitoring and parsing of all logs to analyze user activities such as logon failures and failed attempts to access restricted information.

• Full support for monitoring of virtualized enterprises.

• Alerting interface that generates custom alert actions via email, pager, beep, console message, etc.

• Event correlation modules to constantly monitor for malicious hacking activity. In conjunction with alerts, this is used to inform network security officers and security administrators in real time. This helps minimize the impact of breaches.

• Various types of network activity reports, which can be scheduled or generated as required for any investigation or meeting audit compliances.

• Host-based Intrusion Detection (HIDS).

• Role-based, secure event and reporting console for data analysis.

• Change Monitoring on Windows machines

• USB Tracking, including restricted use, insert/removal recording, and a complete audit trail of all files copied to the removable device.

• Built-in compliance workflows to allow inspection and annotation of the generated reports.

EventTracker is delivered as a software only solution, running on industry standard Microsoft operating systems. It is “virtualization ready” and can be deployed on a single or multiple dedicated or virtual servers. Easy to use, highly scalable and affordable, it represents a solid choice for any organization attempting to meet compliance or attempting to improve their overall IT responsiveness and security.
About Prism Microsystems

Prism Microsystems, Inc. delivers business-critical solutions to consolidate, correlate and detect changes that could impact the performance, availability and security of your IT infrastructure. With a proven history of innovation and leadership, Prism provides easy-to-deploy products and solutions for integrated Security Management, Change Management and Intrusion Detection. EventTracker, Prism's market leading enterprise log management solution, enables commercial enterprises, educational institutions and government organizations to increase the security of their environments and reduce risk to their enterprise. Customers span multiple sectors including financial, communications, scientific, healthcare, banking and consulting.

Prism Microsystems was formed in 1999 and is a privately held corporation with corporate headquarters in the Baltimore-Washington high tech corridor. Research and development facilities are located in both Maryland and India. These facilities have been independently appraised in accordance with the Software Engineering Institute’s Appraisal Framework, and were deemed to meet the goals of SEI Level 3 for CMM.

For additional information, please visit http://www.eventtracker.com/.