Global Information Assurance Certification Paper

Effective Use Case Modeling for Security Information & Event Management

GIAC (GCIH) Gold Certification

Author: Daniel Frye, frizille@gmail.com
Advisor:

Accepted: September 21, 2009

Abstract

In most compliance frameworks and best practices guides there are references to appropriately auditing events within an information technology infrastructure. This places a great deal of importance on appropriately managing event data. However, in recent SANS Advisory Board and SecurityFocus discussions, it is clear that log management is often times an elusive ideal which is near impossible for most companies to implement for a myriad of reasons. Chief among them is the fact that not many organizations truly understand the methods with which to dissect and utilize logging sources. This paper defines a standard methodology which can be used to develop use cases that can be used to help organizations quantify the scope and need for log management technologies.

1. Introduction

With today's technology there exist many methods to subvert an information system which could compromise the confidentiality, integrity, or availability of the resource. Due to the abstract nature of modern computing, the only way to be reliably alerted of a system compromise is by reviewing the system's actions at both the host and network layers and then correlating those two layers to develop a thorough view into the system's actions. In most instances, the computer user often has no indication of the existence of the malicious software and therefore cannot be relied upon to determine if their system is indeed compromised.

To some greater or lesser extent, every system or application has the ability to log its actions. However, the volume of log data generated by systems and the applications running on them is so large that it is impractical for administrators to be able to review every data entry in the log and thus makes the alerting process less than 100% effective. As an example, a single IIS log from a Microsoft Exchange Server running Outlook Web Access for a midsize company with approximately six hundred employees, with its reporting verbosity set to the default verbosity setting as delivered from Microsoft, over a week period generated a daily average of 126MB/day in size comprising of 607,185 lines of data. Without the ability to process this log file in an automated fashion it would take an administrator over 7 days to review the log file if they could review a single line of data every second. This is not very practical nor does it support the ability to review the logs in near real time which is required if actionable intelligence is to be pulled from the log data.

The process of gathering and maintaining network, system, and application log data is commonly referred to using several different definitions. It is sometimes defined as Security Information and Event Management (SIEM), Security Event Management (SEM), Security Information Management (SIM), systems monitoring, and network monitoring (Peikari, Chuvakin, 2004; Contos, Crowell, DeRodeff, Dunkel, Cole,

2007). For the purposes of this paper and to reduce confusion between the practice of Information Security Management (ISM) and its sub-domain of Security Information Management (SIM), all references to the practice of gathering, maintaining, and using log data will be referred to as Security Information and Event Management (SIEM) in this paper.

2. Understanding the Log Management Process

To fully understand SIEM's shortcomings, it is critical to be familiar with the process of log generation, log gathering, and the resulting correlation of log data. The timeline below shows a generic application interaction and the associated log gathering actions which occur after the user's request has been processed.

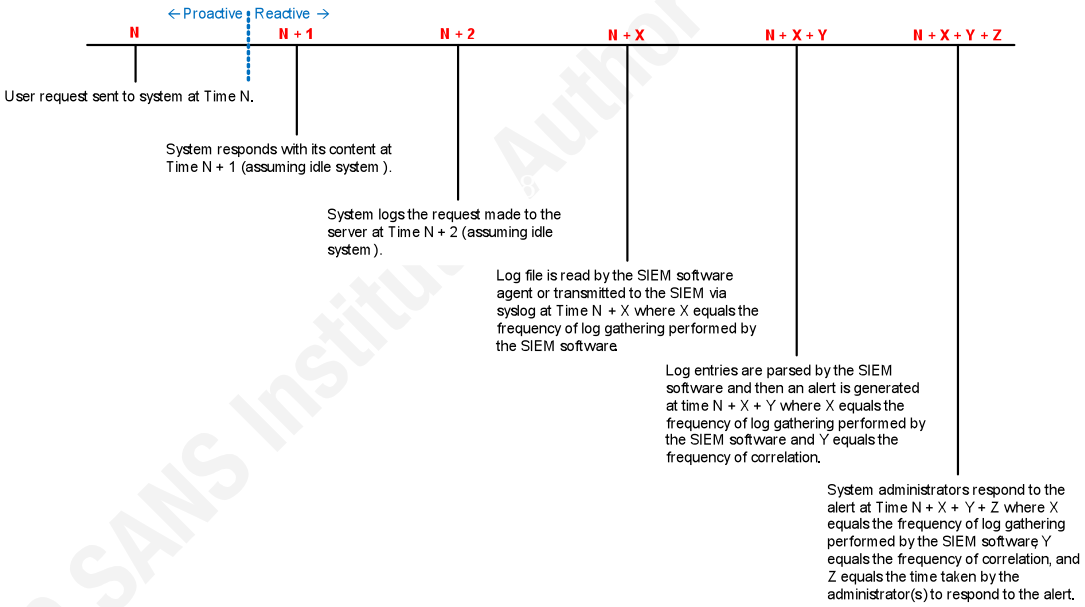

Figure 1

As shown above, there are typically six different actions which comprise the end user request to administrator response process when discussing SIEM.

1. User request
2. System response

3. Write system log
4. Write log file/entry to SIEM system
5. Correlate and analyze SIEM data and generate alert
6. System administrator(s) respond to alerts

An important concept to be aware of is that SIEM data is always *reactive* in nature as the event which is being recorded has already occurred. In security parlance this is termed as being a *detective control*. Depending on the criticality of the event, the times for X, Y, and Z, must be minimized to meet the business requirements for response time.

3. Common Barriers of Successful SIEM Implementations

There are many reasons why SIEM solutions are not the all-in-one magic bullet to an organization's lack of security intelligence.

First and foremost, for SIEM to be truly useful, only actionable data must be sent onward to system and application administrators or security staff. To make SIEM alerts actionable it must address the "Five W's", a basic investigative technique of determining when the event occurred, who was involved, what happened, where did it take place, and why did it happen. The "Five W's" can be mapped directly to common variables in a security investigation.

- When – Time/Date stamp of the event(s) happening
- Who – Identifier of the requestor; typically an IP address and/or a username
- What – Description of the event (such as a GET or POST to a web server)
- Where – System or application that generated the event and where the request originated from
- Why – The purpose of the action and typically is what is being investigated

Despite the benefit of SIEM data, many SIEM system vendors do not quantify the benefit of the use cases they support and therefore do not fully answer the "Five W's" of an investigation. As an example, a popular SIEM platform comes delivered with over 1500 reports out of the box. However, the majority of those predefined use cases are missing fields in their reports, particularly around login events. While their reports may identify when an event occurred, such as a logon failed event (addressing "what"), and which username performed an action (addressing "who"), they do not appropriately identify from which IP address the username attempted to login from and therefore does not adequately answer the "where" question thus making the report not entirely actionable. It may be a case of an employee accessing a server from the corporate office who mistyped their password or it could be a malicious 3rd party attempting to access that host from another host on the server's subnet but with the limited data points you cannot determine if that is the case. The only thing that you can determine is that a login event failed for a particular username a certain number of times but doesn't enable someone to answer the question of "why" the event occurred which during a security incident response scenario is the most important question to answer.

Another common issue with SIEM software is performance. While many SIEM vendors tout tens of thousands of events per second as the capacity of the system, often times that number is misleading. In many cases, each correlation rule subtracts from the overall performance of the system exponentially based on the number of data points being correlated in much the same way that a firewall slows down based on the number of firewall rules in place and the number of filters attached to those particular rules. A firewall with one rule set to filter for a single service is much faster than a firewall set to filter thousands of rules where multiple nested groups exist in each rule for source IP addresses, destination IP addresses, and service. This condition leads to an underestimation of the time needed for correlation as the data set grows and how often those correlations are scheduled. In most SIEM instances correlation is a scheduled task and is not performed as every new event is collected by the SIEM system – if it were the system would conceivably have thousands of

correlations occurring at any single time. As an example, let's assume a bank is interested in monitoring ATM transactions so that security guards can physically apprehend ATM users who are attempting to use stolen card numbers at its banks. The bank has decided to maintain the logging software in a centralized location so that all of its branches can be controlled from one system. Each ATM is configured to send a syslog entry to the central SIEM system with the time/date stamp, the ATM card number, and a location identifier. Under this design, the time for the SIEM system to gather the log data would be *X=0* since the system sends the data in near real time (assuming proper network connectivity to the central location, properly configured ATM's, etc) and can be safely eliminated from calculation. Due to the design of the system, it is conceivable that a correlation would be running for every ATM the bank uses. If the bank had thousands of ATM's, we would be burdening the system with thousands of correlation searches. Due to the correlation load on the SIEM, it takes 300 seconds to search the cardholder database and trigger an alert to the location of the ATM when a match is found thus making our correlation and alert time *Y=300s*. The security guard takes 60 seconds to walk outside and confront the ATM user to make *Z=60s*. The final formula, presented in Figure 1, would then look like this:

$$Time = N + (0) + (300s) + (60s)$$
$$Time = N + (360s)$$

Despite the elaborate system, bank executives wonder why fraudulent ATM use is still occurring. By using the timeline formula from Figure 1, it is apparent that the 300 second correlation time needed by the SIEM system is too great and the ATM user has already completed their transaction and left that bank location well before security is notified.

To further complicate matters, legal regulations and security frameworks require the storage of system log data from one to six years (PCI Security Standards Council, 2009; Federal Register Part II Department of Health and Human Services, 2003).

However, there is very little guidance on exactly what needs to be maintained and how to quantify which log sources truly provide actionable security improvements. These two factors lead organizations towards a path of investing in compliance initiatives without receiving the benefits of investing in security initiatives (Rohmeyer, 2009).

Another misconception is that event data from a SIEM will be useful "out of the box". In fact, it cannot be further from the truth. As an example, on a Linux system there are many different logging levels (Welsh, Dalheimer, Kaufman, 1999). Without properly tuning systems to send the relevant data to the SIEM software it will not be possible to gather and report on the data. A common euphemism is "garbage in, garbage out" when dealing with collecting event data. It is critical that an organization catalog the levels of logging needed to meet their SIEM goals.

Along the same lines as validating the types of data being sent to the SIEM solution, an organization must understand what business use cases those data sources are supporting. As previously discussed, sending too much data to a SIEM system will burden it with correlating and processing data unnecessarily, thus leading to poor performance which will ultimately affect the usefulness of the SIEM. In conjunction to increasing the difficulty of achieving the goals of the SIEM, ultimately the amount of labor spent configuring the logging sources unnecessarily is wasted labor and will negatively affect the return on investment of the SIEM solution.

4. Method for Developing Use Cases

When discussing SIEM it is important to understand the difference between a business use case and a system use case. A business use case is a general business requirement, such as "identify failed login events", which is being addressed by the SIEM solution (Evans, 2004). A system use case is a specific technology component in the SIEM system itself, such as "alert on a failed Windows login event", and specifies explicitly what data the system is

manipulating (Evans, 2004). In many SIEM vendor brochures they do not distinguish between a business use case and the more detailed system uses cases for their SIEM. When identifying needs for a SIEM and in the future evaluation, purchasing, and implementation phases, it is important to keep the use case definitions to the most explicit form possible and develop system use cases.

As an example, assume there is a use case referred to as "identify failed login events" by a vendor. The vendor claims to provide a return on investment from eliminating risk to the organization by alerting on repeated failed login events. This by itself is not easily quantifiable and therefore does not make a good metric for return on investment calculation and is poorly advertised as such (Jacquith, 2007).

Conversely, by narrowing the scope of the ambiguous use case "identify failed login events" to a direct system relationship it is possible to demonstrate some form of return on investment and risk mitigation. A truly useful metric can be efficiently and repeatedly measured with a high degree of accuracy (Jacquith, 2007). As an example, if a company had two hundred Windows servers and system event log monitoring was deployed on one hundred of its Windows hosts, it would be possible to quantify that 50% of the company's servers were being actively monitored for failed login events. While the risk of a brute force attack is still present, the metric that the company is monitoring 50% of its environment for such an attack can be used to accurately identify the organizations coverage of its monitoring control. Let's assume in the following six months the company deploys monitoring to its remaining servers so it now has 100% monitoring coverage for failed login events. The increase in the coverage metric from 50% to 100% can be shown to executive level management to demonstrate that the security team is improving the processes within the organization and thereby reducing risk, although the exact return on investment (ROI) is still an abstract quantity.

Use case selection for an event logging solution depends entirely on the business goal for the tool. While there are many business goals for which event log analysis is useful this paper focuses on using an event logging tool for security purposes.

As part of obtaining the Certified Information Systems Security Professional (CISSP) certification, security professionals are trained to protect information from three main threat categories; Confidentiality, Integrity, and Availability (Bragg, 2002). These are commonly referred to as the CIA triad of information security (Bragg, 2002).

In terms of information security incident response, security professionals should expect to be called upon to investigate incidents from any of the three CIA triad categories. As such, the development of uses cases to track activities on information resources should be done in a manner that aids in investigating those types of incidents.

To enable the development of uses cases a thorough and detailed analysis of the data available to an organization must be done prior to selection of a SIEM solution.

4.1 Dissecting Use Cases

The proposed approach to SIEM design is the Top-Down Bottom-Up Middle-Out approach abbreviated as TDBUMO (pronounced *teddy-bumo*). Through this design method a business problem is defined into how it can be accomplished by starting at the top, the bottom, then working from the shortest path found in the middle to the outermost edges.

To demonstrate using the TDBUMO process for SIEM design, let's assume an organization hosts a popular Enterprise Resource Planning (ERP) application for its customers, and as such has a considerably large and diverse

technology base for which it is responsible. To be compliant with HIPAA and PCI the organization has determined it needs to deploy a system to review and correlate its system logs. The organization has no existing logging processes in place currently.

4.1.1 Top Down

In the Top Down view of the SIEM solution, the goal is to understand how data will flow into the system itself. If the SIEM design is approached by viewing it as a tree graph, the SIEM solution would be the root node. From there, a very high level view can be broken into multiple categories by grouping like systems based on how data will flow into it. In the example of the ERP hosting company, three different categories can be used; Windows, Unix, and Network. Pictorially, the organization can represent their feeds into the SIEM cloud, which remains abstract since the design is vendor independent at this stage, as shown in Figure 2.

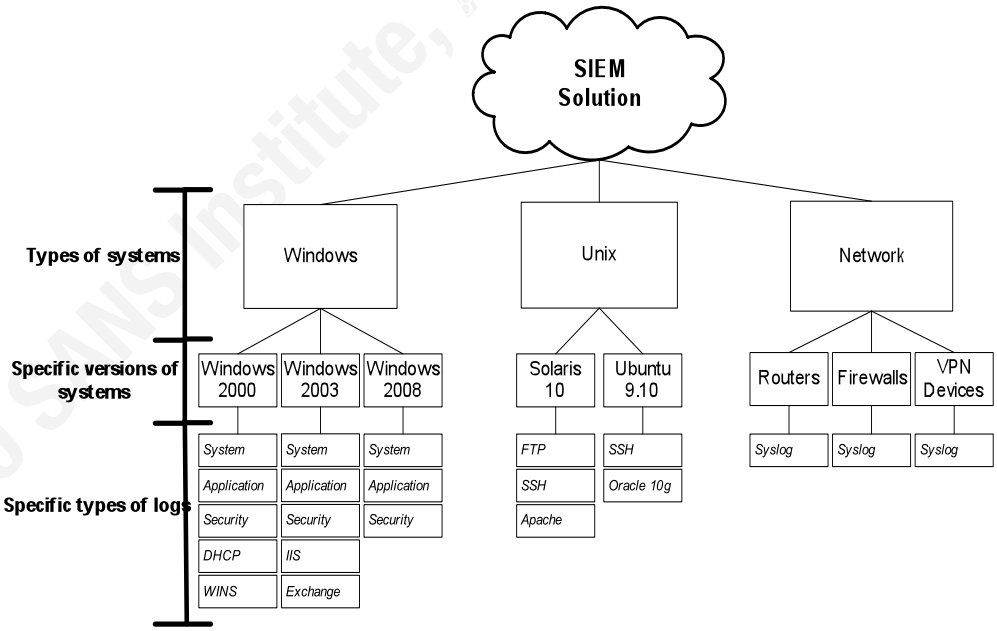

Figure 2

As Figure 2 shows, similar systems have been grouped together into a manageable number of categories. From here, it is possible to reduce each division further into its unique systems until the tree graph has been filled out with the correct number of leaves where each leaf is the different type of logs that exist on that system. Note: Beware of the natural tendency to group similar systems in the first few levels of the tree. Even though some systems may be very similar, such as a Windows 2000 Server and Windows 2003 Server, it is critical to separate them into different categories at this stage. As an example, Windows 2000 does not support WMI calls for event log collection while a Windows 2003 system does. Grouping these systems can cause design and collection issues later during a SIEM deployment as their collection methods will then be incompatible. Likewise, the Windows and Unix based systems were separated since an agent is typically designed to run on either Windows or UNIX, but not both.

Once a pictorial model is made of how data will flow into the SIEM solution transport methods must be assigned to each of them. In this case, log files reside on the Windows and UNIX systems which must be read and sent to the SIEM solution. Typically, WMI calls will be used on the Windows servers and a combination of agent or syslog feed will be used for the UNIX servers. For the Network devices the standard syslog output will be used.

Once the transport method has been determined, as well as the associated network port requirements, it is important to review the network topology in place where the servers reside. If there are firewalls and/or DMZ's between the SIEM solution and the servers, which is usually the case, oftentimes collection methods will need to be validated. As an example, the basic WMI implementation found in Windows 2003 and Windows 2008 Servers utilizes a random high port (greater than 1024 TCP) for its communication (Nelson, 1998). This in effect requires that internal firewalls allow certain ranges of high ports from the SIEM solution to other servers on the network and

creates a security paradox as firewall rules must be opened, and therefore the security posture relaxed, to gather the data. Conversely, this can be solved by running a vendor specific WMI service or utilizing an agent to push the logs to the SIEM solution.

4.1.2 Bottom Up

The Bottom Up analysis starts at the bottom of the pictorial graph identified in Figure 2 and works its way up. As shown in Figure 2, the lowest leaves on the graph are the specific log types found in the environment. For example, the organization knows that its Windows 2003 servers have a combination of System, Application, Security, IIS, and Exchange logs present on them. Not every server will have all of those logs, but when looking at any server running Windows 2003, the organization can check to validate that each of those logs is being looked for. More specifically, it is possible to do an analysis on exactly what data points are in those log types.

Each piece of information in a log file, such as an IP Address, is termed a *data point*. Every log file is made up of one or more data points. These data points provide the backbone for event correlation. For example, in the Windows 2003 IIS logs, the IP Address of the connecting user is recorded in one of the fields. Similarly, in the Apache logs found on the Solaris 10 servers, an IP Address field is also present. When two data points contain the same type of data, this is termed as a *collision*. Collisions are the basis of correlation in that when an event matches in the IIS logs, it may also be detected in the Apache logs. Conversely, a username and an IP Address would not be a collision as they are different data points regardless of the log source.

During the Bottom Up analysis, the organization implementing the SIEM must catalog the different characteristics of its log files. Several useful characteristics are listed in the table below.

Log file name	
Log file description	
Log file location	
Log verbosity level	
Log size average	
Log type	
Data points	
Application name	
Application version	
Operating System	
Rotation frequency	Real-time, hourly, daily, etc
Log standard defined	Link to the operating procedure

Once all the different types of log files are categorized, it is necessary to determine what data points are in them. For this step, vendor documentation can be used to determine exactly how the log files are comprised and written. The specific data points found in each log will serve as the basis for use case analysis in the Middle-Out section of this paper.

When categorizing the log structures the log verbosity level is particularly important. As discussed previously, many applications log certain levels of detail based on their verbosity settings. Having too much or too little detail in the application log file can be as detrimental to running a SIEM effectively as having no log at all. It is important to perform an analysis at the most verbose setting for the application then gradually back down the setting until only the required data points are present. In most cases, some junk data points will still be present but by eliminating many of them prior to collection, the SIEM solution, regardless of vendor, will run much more efficiently and the resources required to run it will be much smaller.

During SIEM vendor selection, it is very important to analyze how the SIEM solution will read in the particular log types. Not all log parsers are created equal and in many instances changes to the log formats occur when new versions of applications are released. This will result in the SIEM solutions parsing patterns to no longer match the data being retrieved and in effect the SIEM solution's correlation ability has been marginalized or at least reduced. For this reason it is critical that an SIEM solution maintain a view into the unparsed and originally retrieved log file. SIEM parsing cannot always be relied upon to adequately normalize and analyze log data for this reason, however, the un-normalized and unmodified text strings being captured can be searched and analyzed if the SIEM operators understand which data points are present in those logs.

4.1.3 Middle Out

In the Middle Out portion of the design process, the goal is to take the data points determined in the Bottom Up phase and match them to use cases across the different systems found in the enterprise. By looking at the complete list of data points, it is very easy to construct use cases from the data to support an objective. For example, let's assume the ERP hosting company wants to track failed login attempts. By referencing the data from the Bottom Up portion of the design, all of the locations of username and IP Address can be determined regardless of the system or log type. From there, it is a relatively simple exercise to write a correlation rule in the SIEM solution for all the identified data sources.

During use case analysis, it is of value to track all of the uses cases developed according to the following fields.

Use case name	
Use case description	

Author Name, email@address

Data points required	
Importance	
Required speed	
Alert Method	
Alert output format	

There are many methods by which to analyze and develop use cases. The most effective methods will vary from organization to organization but a starting point for all organizations is the same – asking the question what is important to maintain a profitable business model and to reduce risk to that model?

In all instances use cases will fall into one of the three CIA Triad categories – Confidentiality, Integrity, or Availability. In the case of the ERP hosting company used as an example above, all three of the categories are important as all of them reflect a significant business risk whether it be a data breach, employees modifying data without authorization, or system outages which would affect a customer's service level agreement and thus expose the business to unneeded financial losses.

To develop the proper use cases a strong partnership with the individual business units must be forged. It is important to let the business units have some stake in the SIEM project as it is their information which is being protected. Business unit buy-in to the SIEM solution will also help with acquiring the needed capital expenditure dollar amounts to make an implementation successful. Underfunding a SIEM deployment is almost as detrimental as not having the correct log analysis performed.

Author Name, email@address

5. Conclusion

The implementation of SIEM solutions to collect, store, and make efficient use of event log data to meet security and compliance goals is a critical objective for almost every security team, especially those with regulations which stipulate log review as a required component of their security program. Despite the many challenges of implementing a SIEM solution, an organization can be successful if the organization understands how the SIEM solution will communicate and gather data, what data is available in the logs it keeps, and a detailed understanding of what use cases best make up the reason for the substantial investment in an SIEM solution.

The Top-Down Bottom-Up Middle-Out (TDBUMO) process of SIEM design provides the necessary information to any organization implementing any SIEM solution. In many instances, organizations go through multiple SIEM deployments to arrive at the same level of understanding as the TDBUMO process provides but with significantly more effort, more financial investment, and more resource utilization. As mentioned previously, "garbage in, garbage out" is often used to describe SIEM correlation. As it is with correlation, poor analysis and poor planning will only result in poor SIEM coverage and performance, i.e., "garbage in, garbage out".

References

Jaquith, Andrew. (2007). *Security Metrics*. Upper Saddle River, NJ: Addison-Wesley Professional.

Peikari, Cyrus & Chuvakin, Anton. (2004). *Security Warrior*. Sebastopol, CA: O'Rielly and Associates.

Welsh, Matt, Dalheimer, Matthias Kalle, & Kaufman, Lar. (1999). *Running Linux, Third Edition*. Sebastopol, CA: O'Rielly and Associates.

Author Name, email@address

Contos, Brian T., Crowell, William P., DeRodeff, Colby, Dunkel, Dan, & Cole, Eric. (2007). *Physical and Logical Security Convergence*. Burlington, MA: Syngress Publishing, Inc.

Strunk, William Jr. & White, E. B. *The Elements of Style*. New York, NY: Longman

Bragg, Roberta. (2002). *Cissp - certified information systems security professional*. Que Publishing.

Federal Register Part II Department of Health and Human Services, Office of the Secretary 45 CFR Parts 160, 162, and 164. *Health Insurance Reform: Security Standards; Final Rule.* (2003, February). Retrieved from http://aspe.hhs.gov/admnsimp/FINAL/FR03-8334.pdf

PCI Security Standards Council. (2009, July). *Payment Card Industry Data Security Standard Version 1.2.1*, Retrieved from https://www.pcisecuritystandards.org/security_standards/download.html?id=pci_dss_v1-2.doc

Evans, G. (2004, July). *Getting from use cases to code, part 1: use-case analysis*. Retrieved from http://www.ibm.com/developerworks/rational/library/5383.html

Rohmeyer, P. (2009, November). *Standards compliance does not equal sound information security risk management*. Retrieved from http://searchsecurity.techtarget.com/magazineFeature/0,296894,sid14_gci1373558_mem1,00.html

Nelson, M. (1998, June). *Using Distributed COM with Firewalls*. Retrieved from http://msdn.microsoft.com/en-us/library/ms809327.aspx

Author Name, email@address