Understanding and Selecting SIEM/Log Management

Version 2.0
Released: August 25, 2010

Securosis, L.L.C. 20930 North Tatum Blvd Suite #110-116 Phoenix, AZ 85050 **T** 602-412-3051 info@securosis.com www.securosis.com

Author's Note

The content in this report was developed independently of any sponsors. It is based on material originally posted on the Securosis blog <http://securosis.com>, but has been enhanced, reviewed, and professionally edited.

Special thanks to Chris Pepper for editing and content support.

Licensed by Nitro Security

NitroSecurity is the leader in high-performance, content-aware security information and event management solutions. NitroSecurity's integrated NitroView solution provides "single pane of glass" visibility into events and logs and monitors network, database and application payload information. Utilizing the industry's fastest analytical tools, NitroView identifies, correlates, and remediates threats in minutes instead of hours, making organizations more secure and efficient. For more information, please visit http://www.nitrosecurity.com.

Contributors

The following individuals contributed significantly to this report through comments on the Securosis blog and follow-on review and conversations:

Dr. Anton Chuvakin
Brian Jones
Steve Lafferty
'Mark'
Dwayne Melancon
Michael Leland
Chris Poulin
Ed Rarick
Maceo D. Wattley, M.A.

Copyright

Table of Contents

Introduction

Over the past decade business processes have been changing rapidly. We focus on collaborating more effectively, both inside and outside our own organizations. We have to support more devices in different form factors, many of which IT doesn't directly control. We add new applications on a monthly basis, and are currently witnessing the decomposition of monolithic applications into dozens of smaller loosely connected application stacks. We add virtualization technologies and SaaS for increased efficiency. Now we are expected to provide anywhere access while maintaining accountability, but we have less control. A lot less control.

If that wasn't enough, bad things are happening much faster. Not only are our businesses always on, the attackers don't take breaks, **ever**. New exploits are discovered, 'weaponized', and distributed to the world within hours. So we need to be constantly vigilant and we don't have much time to figure out what's under attack and how to protect ourselves before the damage is done. Compound these 24/7 demands with the addition of new devices implemented to deal with new threats. Every device, service, and application streams zillions of log files, events, and alerts. Our regulators now mandate we analyze this data every day. But that's not the real issue.

The real issue is pretty straightforward: of all the things flashing at us every minute, we don't know what is really important. *We have too much data, but not enough information.*

This lack of information compounds the process of preparing for the inevitable audit(s), which takes way too much time for folks who really should be dealing with security issues. Sure, most people just bludgeon their auditors with reams of data, none of which provides context or substantiation for the control sets in place relative to the regulations in play. But that's a bad answer for both sides. Audits take too long and security teams never look as good as they should because they generally can't prove what they are doing.

Ask any security practitioner about their holy grail and the answer is twofold: They want one alert specifying exactly what is broken, on just the relevant events, with the ability to learn the extent of the damage. They need to pare down billions of events into actionable information.

Second, they want to make the auditor go away as quickly and painlessly as possible, which requires them to streamline both the preparation and presentation aspects of the audit process.

Security Information and Event Management (SIEM) and Log Management tools have emerged to address these needs and continue to generate a tremendous amount of interest in the market, given the compelling use cases for the technologies.

Defining SIEM and Log Management

Security Information and Event Management (SIEM) tools emerged about 10 years ago as the great hope of security folks constantly trying to reduce the chatter from their firewalls and IPS devices. Historically, SIEM consisted of two distinct offerings: SEM (security event management), which collected, aggregated and acted upon security events; and SIM (security information management), which correlated, normalized and reported on the collected security event data.

These days integrated SIEM platforms provide near real-time monitoring of network and security devices, with the idea of identifying the root causes of security incidents and collecting useful data for compliance reporting. Most end users believe the technology is at best a hassle and at worst an abject failure. SIEM is widely regarded as too complex, and too slow to implement, without providing enough customer value to justify the investment.

While SIM & SEM products focused on aggregation and analysis of security information, Log Management platforms were designed within a broader context of the collection and management of any and all log files. Log Management solutions don't have the negative perception of SIEM because they do what they say they do: aggregate, parse, and index logs.

Log Management has helped get logs under control, but under-delivered on the opportunity to derive value from the archives. Once again: more data, less information. Collection, aggregation, and reporting are enough to check the compliance box, but not enough to impact security operations — which is what organizations are really looking for. End users want simple solutions that improve security operations, while also checking the compliance box.

Given that backdrop, it's clear the user requirements that were served by separate SIEM and Log Management solutions have fused. And so these historically disparate product categories have fused as well. If not from an integrated architecture standpoint, then certainly from the standpoint of user experience, management console, and value proposition. There really aren't independent SIEM and Log Management markets any more.

The key features we see in SIEM/Log Management solutions include:

- **Log Aggregation** — Collection and aggregation of log records from the network, security, servers, databases, identity systems, and applications.
- **Correlation** — Attack identification by analyzing multiple data sets from multiple devices to identify patterns not obvious when looking at only one data source.
- **Alerting** — Defining rules and thresholds to display console alerts based on customer-defined prioritization of risk and/or asset value.
- **Dashboards** — An interface which presents key security indicators to identify problem areas and facilitate investigation.
- **Forensics** — The ability to investigate incidents by indexing and searching relevant events.
- **Reporting** — Documentation of control sets and other relevant security operations and compliance activities.

In this report, Securosis spotlights both the grim realities and real benefits of SIEM/Log Management platforms. The vendors are certainly not going to tell you about the bad stuff in their products, instead shouting out the same fantastic advantages touted in the latest quadrant report. Trust us when we say there are many pissed-off SIEM users, but there are plenty of happy ones as well. We want to reset expectations to make sure you focus on success and avoid joining the former category.

Use Cases

When you think about it, security success in today's environment comes down to a handful of key imperatives. First we need to *improve the security of our environment*. We are losing ground to the bad guys, and we've got to make some inroads on more quickly figuring out what's being attacked and protecting it.

Next we've got to *do more with less*. Yes, it seems the global economy is improving, but we can't expect to get back to the halcyon days of spend first, ask questions later — ever. With more systems under management we have more to worry about and less time to spend poring over reports, looking for the proverbial needle in the haystack. Given the number of new attacks — counted by any metric you like — we need to increase the efficiency of resource utilization.

Finally, auditors show up a few times a year, and they want their reports. Summary reports, detail reports, and reports that validate other reports. The entire auditor dance focuses on convincing the audit team that you have the proper security controls implemented and effective. That involves a tremendous amount of data gathering, analysis, and reporting to set up — with continued tweaking required over time. It's basically a full time job to get ready for the audit, dropped on folks who already have full time jobs. So we must *automate those compliance functions* to the greatest degree possible.

Yes, there are many other reasons organizations embrace SIEM and Log Management technology, but these three make up the vast majority of the projects we see funded. So let's dig into each use case and understand exactly what problem we are trying to solve.

Use Case #1: React Faster

Consider the typical day of a security analyst. They sit down at their desk, check out their monitors, and start seeing events scroll past. A lot of events, probably millions. Their job is to look at that information, figure out what's wrong, and identify the root cause of each problem.

They probably have alerts set up to report critical issues within their individual system consoles, in an effort to cull down the millions of events into some finite set of things to investigate — per system. So the analyst goes back and forth between the firewall, IPS, and network traffic analysis consoles. If a Web Application Firewall or a Database Activity Monitoring product is deployed alongside these other tools, the analyst needs to deal with those as well. An office chair that swivels easily is a good investment to keep the analyst's neck from wearing out and minimize trips to the chiropractor.

Security analysts tend to be pretty talented folks, so they do find stuff based on their understanding of the networks and devices and their own familiarity with normal activity, which allows them to recognize 'not normal'. There are some events that just look weird but cannot be captured in a policy or rule. Successful detection is contingent on the ability of the human analyst to interpret the alerts between the various systems and identify attacks.

The issues with this scenario are numerous:

- **Too much data, not enough information** — With anywhere from 10 to 2,000 devices to monitor, each generating thousands of log events and/or alerts a day, there is plenty of data. The analyst must turn that data into information, which is a tall order for anyone.
- **High signal to noise ratio** — With so much data, the analyst is likely only going to investigate the most obvious attacks. And without some way to reduce the number of alerts to deal with, there will be plenty of false positives to wade through, impacting productivity, and reducing effective coverage.
- **No "situational awareness"** — The new new term in security circles is *situational awareness*; the concept that anomalous situations are lost in a sea of detail unless the larger business context in considered. With only events to wade through, a human analyst will lose context and not be able to keep track of the big picture.
- **Too many tools to isolate root cause** — Without centralizing data from multiple systems, there is no way to know if an IPS alert was related to a web attack or an attempt to exfiltrate intellectual property. So the analyst needs to quickly move from system to system to validate and confirm the attack, and to understand the depth of the issue. That approach isn't efficient, and in an incident response situation, time is the enemy.

We've written on numerous occasions about the need to *react faster*, because we can't predict where the next attack is coming from. The promise of SIEM and Log Management solutions is to help us react faster — and better — and make the world a safer place, right? The features and functions a security analyst will employ in this case are:

- **Data aggregation** — SIEM/LM solutions aggregate data from many sources, including networks, security, servers, databases, applications, etc. — providing the ability to monitor everything. Having all the events in one place helps avoid missing anything.
- **Correlation** — Correlation looks for common attributes, and links events together into meaningful bundles. Being able to look at all events during a particular window, or everything a specific user did, gives us a meaningful way to investigate security events. This technology provides the ability to perform a variety of correlation techniques to integrate different sources, in order to turn data into useful information.
- **Alerting** — Automated analysis of correlated events can produce more substantial and detailed alerts, and help identify what needs to be investigated right now.
- **Dashboards** — With liberal use of eye candy, SIEM/LM tools take event data and turn it into fancy charts. These charts can assist the analyst in seeing patterns, and more importantly in seeing activity that deviates from standard patterns, which is not visible when looking at individual log entries.

So ultimately this use case provides the security analyst with a set of automatic eyes and ears to wade through all the data and help identify what's most important and requires attention now.

Use Case #2: Improve Efficiency

Turn back the clock to your last budgeting cycle — you weren't surprised when you got the news that budgets are going down again, were you? At least they didn't make you cut staff during the "right-sizing" at the end of 2008, eh? Of course, budget and resources be damned, you are still on the hook to secure the new applications, which will require some new security gadgets which generate more data. The auditors don't say, "Hey, times are tough out there, don't worry about protecting that data," do they? Didn't think we would get that lucky.

We cannot afford to forget the audit deficiencies detailed in your friendly neighborhood assessor's last findings. Yes, those have to be dealt with too, and some time in the next quarter, because the audit is scheduled within three months. This may seem like an untenable situation but it's all too common. Security professionals must continue looking for opportunities to improve efficiency and do more with less.

As we look deeper into this scenario, there are a couple inevitable situations we must deal with:

- **Compliance requirements** — Government and industry regulations force us to demonstrate compliance — which requires gathering log files, parsing and discarding unneeded events, and analyzing transactions into human-readable reports to prove you're doing things correctly. IT and Security must help Audit determine which events are meaningful, so regulatory controls are based on complete and accurate information, and internal and external audit teams define how this data is presented.

- **Nothing gets shut down** — No matter how hard we try, we cannot shut down old security devices that protect a small portion of the environment. So every new device and widget increases the total amount of resources required to keep the environment operational. And the threats they protect us from may not be our first concern, but they have not gone away, and still require attention. Given the number of new attack vectors clamoring for new protection mechanisms, this problem is going to get worse.

- **Cost center reality** — Security is still an overhead function and as such, it's expected to work as efficiently as possible. That means no matter what the demands, there will always be pressure to cut costs.

So this use case is all about how SIEM/LM can improve efficiency of existing staff, allowing them to manage more devices which are detecting more attacks, all while reducing the time from detection to remediation. A tall order, sure, but let's look at the capabilities we have to accomplish this:

- **Data aggregation** — Similar to our 'react faster' use case, having consolidated access to more data means less time is wasted moving between systems (swivel chair management). This increases efficiency and should allow security analysts to support more devices.

- **Dashboards** — A picture is worth a thousand words, so a well architected security dashboard has to be worth more than that. When trying to support an increasing number of systems, the ability to see what's happening and gain context with an overview of the big picture is critical.

- **Alerts** — When your folks need to increase their efficiency, they don't have a lot of time to waste chasing down false positives and investigating dead ends. So having the ability to fire off alerts based on real events rather than gut feel saves everyone a lot of time.

- **Forensic investigations** — Once the problem is verified, it becomes about finding the root cause as quickly as possible and then understanding the extent of the damage. The SIEM/LM solution can provide the context and information needed to determine the nature of the attack and to what degree it's proliferated — all critical information to containing the damage of a breach. It's about working smarter, not harder.

- **Automated policy implementation** — Some SIEM/LM tools can build automated policies based on observed traffic. This baseline (assuming it represents normal and healthy traffic) enables the system to start looking for *not normal* activity much faster, which may warrant further investigation.

This use case is really about doing more with what you already have, which has been demanded of security professionals for years. There has been no lack of tools and products to solve problems, but the resources and expertise to take full advantage of those capabilities can be elusive. Without a heavy dose of automation, and most importantly a significant investment to get the SIEM/LM system configured appropriately, there is no way we can keep up with the bad guys.

Use Case #3: Compliance Automation

You know the feeling you get when you look at your monthly calendar, and it shows an upcoming audit? Whatever you were planning to do goes out the window, as you spend countless hours assembling data, massaging it, putting it into fancy checklists and pie charts, and getting ready for the visit from the auditor.

Some organizations have folks who just focus on documenting security controls, but that probably isn't you. So you've got to take time from the more strategic, or even operational, tasks you've been working on to prepare for the audit. And it gets worse, because every regulation has its own vernacular and rule set — even though they are talking about the same sets of security controls. So there is little you can leverage from last month's PCI audit to help prepare for next month's HIPAA assessment.

And don't forget that compliance is not just about technology. There are underlying business processes in play that can put private data at risk, which have to be documented and substantiated as well. This requires more domain expertise than any one person or team possesses. The need to collaborate on a mixture of technical and non-technical tasks makes preparing for an audit that much harder and more resource intensive.

Also keep in mind the opportunity cost of getting ready for audits. For one, time spent in Excel and PowerPoint massaging data is time you aren't working on protecting information or singing the praises of your security program. And managing huge data sets for multi-national organizations, perhaps across hundreds of sites, requires ninja-level Microsoft Office skills. Drat, don't have that.

As if things weren't hard enough, regulatory audits tend to be more subjective than objective, which means your auditor's opinion can make the difference between the rubber stamp and a book of audit deficiencies that will keep your team busy for two years. So getting as detailed as possible and substantiating your interpretations of the regulations with data helps make your case. And providing that data takes time. Right — time you don't have.

So this scenario focuses on the need to automate compliance, provide mechanisms to automate preparation to the greatest degree possible, and standardize the formats of reports based on what works. We are trying to move from many audits and many redundant preparations, to one control and one report supporting many regulations and audits.

The features in most SIEM/LM sets to address this situation are:

- **Data aggregation** — Once again, having centralized access to data from many devices and computing platforms dramatically reduces the need to manually gather information, and lets you start focusing on analysis as quickly as possible.

- **Pre-built compliance reports & polices** — Of course, you aren't the only company dealing with PCI, so the vendors have built reports for the leading regulations directly into their products. To be clear, it's not like you can hit a button and make the auditor go away. But you at least have a place to start with data types mapped to specific regulations.

- **Secure archival of events** — Substantiation is all about the opinion of the auditor and your ability to convince him/her that the controls are in place and effective. Having an archive of relevant events and other analysis provides a means to use *data* (as opposed to speculation) to prove your point.

- **Workflow and collaboration with SoD** — Compliance reporting is a process which requires management and collaboration. SIEM/LM tools generally have some simple workflow built in to track who is doing what, and make sure folks don't step on each other's toes during preparation. The SIEM/LM also helps enforce separation of duties (SoD) to ensure there is no question of the integrity of the reporting.

From on what we see, most SIEM/LM projects aim to address one of these three scenarios. But knowing what problem you are trying to solve is only the first requirement before you can select a product. You need to get everyone else on board with the decision, and that requires business justification, which is our next topic.

Business Justification

We have discussed the use cases that are driving organizations to consider SIEM and Log Management products, but those don't get projects funded. Next we need to focus on making the business case for the project and examine how to justify the investment in bean counter lingo.

End User Motivations and Business Justification

Securosis has done a lot of work on isolating the motivation for security investments. Unfortunately our research shows budgets are allocated to visceral security issues people can see and feel, rather than based on critical consideration of risks to the organization. In other words, it's much harder to get the CEO to sign off on a six-figure investment when you can't directly demonstrate a corresponding drop in profit or an asset loss. Complicating matters is the fact that in many cases, such as credit card theft, someone else suffers the losses. So compliance and/or regulation is really the only way to justify investment to address the quiet threats.

The good news for SIEM and Log Management is that the technology is really about improving efficiency by enhancing our ability to deal with the mushrooming quantities of data generated by network and security devices. Or being able to detect an attack designed to elude a firewall or IPS. Or even making reporting and documentation (for compliance purposes) more efficient. You can build a model to show improved efficiency, so of all security technologies — you'd figure SIEM/Log Management is relatively straightforward to justify.

Of course, putting together a compelling business justification does not always result in a funded project. When money gets tight (and when is money not tight?) sometimes it's easier to flog employees to work harder, as opposed to throwing high-dollar technology at the problem. Automation is good, but quantifying the real benefits can be challenging.

Going Back to the Well

Our efforts are hamstrung by a decade of mismatched expectations regarding security spending. Our finance teams have seen it all, and in many cases haven't seen the tangible value of the security technology. So they are justifiably skeptical about yet another ROI model showing a two week payback on a multi-million dollar investment. Okay, that's a bit facetious, but only a little.

When justifying any investment we need to make sure not to get hung up on measuring what can't be accurately measured, which inevitably causes the model to collapse under its own cumbersome and unrealistic assumptions. We also need to move beyond purely qualitative reasoning, which produces hard to defend results. Remember that security is an investment that produces neither revenue nor fully quantifiable results, so trying to model it precisely is asking for failure.

Ultimately, having both bought and sold security technologies for many years, we have concluded that end user motivations can be broken down into two buckets: Save Money and Make Money. Any business justification needs to

very clearly show the bean counters how the investment will either add to the top line or help improve the bottom line. And that argument is far more powerful than eliminating some shadowy threat that may or may not materialize.

Of course, in some industries implementing some form of log management is not optional. There are regulations such as PCI that specifically call out the need to aggregate, parse, and analyze log files. So the point of justification becomes what kind of infrastructure is needed at what level of investment since solutions range from free to millions of dollars. To understand where our economic levers are as we build the justification model, get back to the use cases, and show how these can justify the SIEM/Log Management investments. We'll start with the two use cases that are straightforward to justify because each involves hard cost reduction.

Compliance Automation

The reality is that most SIEM/Log Management projects are funded from the compliance budget, because compliance mandates are not optional. For example, if your board of directors mandates new Sarbanes-Oxley controls (they generally agree on not going to jail), you are going to implement them. If your business accepts credit cards on Internet transactions, you are going to comply with the PCI data security standard. Compliance automation is a "must do" business justification because regulatory or compliance requirements *must* be met.

But how do you justify a tool to make the compliance process more efficient? Get out your stopwatch and start tracking the time it takes you to prepare for these audits. Odds are you know how long it took to get ready for your last audit, and next time the auditor is going to look over your shoulder more — asking for additional documentation on policies, processes, controls, and changes to the stuff you did last time around.

This business case is based on the amount of time it takes to prepare for each audit, which is going to continue going up, and the need for automation to keep those costs under control. Gathering, managing, analyzing, and reporting on events is complex, and throwing more people at the job is not cost effective. Whether the audit preparation budget gets allocated for people or tools shouldn't matter. The good news is you pay for SIEM/Log Management with the compliance budget, but its value accrues to security too, and streamlines operations. Sounds like a win/win to us.

Operational Efficiency

Our next use case is about improving efficiency, and this is relatively easy to justify. If you look back at the past few years, the perimeter defenses of your organization have expanded significantly. This perimeter sprawl is due to purpose-built devices implemented to address specific attack vectors. Think email gateway, web filter, SSL VPN, application aware firewall, web application firewall, etc. Each of these has a legitimate place in a strong perimeter. But each device requires management to set policies, monitor activity, and respond to potential attacks. Each system requires time to learn, time to manage, and time to update. All of which requires people, and additional people aren't really in the spending plan nowadays.

Operational efficiency means less time managing, while accomplishing the security checklist items you are responsible for. There is clearly a cost to having analysts gather data from numerous security consoles and do analysis and correlation in their own heads. Automating much of this analysis clearly increases the efficiency and the effectiveness of the folks you already have.

So you can justify this use case by talking about the number of staff you won't have to hire thanks to intelligent use of automation. Lots of folks are scared by this justification, because they figure talking about efficiency means they'll be expected to cut headcount. That's short-sighted, because most security teams are woefully understaffed, which means

their only hope is to automate activities and improve efficiency. Using time and cost metrics for these tasks, you can directly generate a number for activities that you won't have to perform manually, thus making better use of resources and saving money.

React Faster

Finally, the 'react faster' use case focuses on improving security by accelerating detection of attacks. So how do we quantify the benefits of this kind of use case? Let's look at the economic impact of improved security:

- **Reduced downtime (increased availability)** — Here we'd need to model the cost of downtime by some credible metric. Perhaps lost orders, unhappy customers, or failure to ship merchandise. There are many ways to quantify downtime — odds are your operations folks have a method they've used to justify their management tools. There is no need to reinvent the wheel.

- **Brand impact of breach** — You can also try to estimate the negative effect on the organization's brand of a high profile breach. Right, that's about as scientific as throwing a dart.

- **Cost savings on disclosure and clean-up** — Less squishy is the cost of notifying customers of a breach and cleaning up the mess caused by a successful attack. There are standard numbers for these costs based on number of customer records compromised, so this is pretty easy to quantify.

Clearly this is the hardest situation to justify compellingly. Securosis has done a lot of research into this kind of squishy business justification, and tried to make it a bit less squishy. We recommend you take a look at our Business Justification for Data Security white paper to get a feel for how to structure a process to quantify risks and losses when hard numbers are not available (http://securosis.com/blog/the-business-justification-for-data-security--version-1.0/). Sure, we think ROI is the flying unicorn of statistics, but we do provide qualitative and quantitative approaches for producing estimates of tangible cost reduction. Tools provide a better likelihood of detecting attacks, and a lot more information for understanding how to address problems.

To be clear, cost-based justification is not easy. And we are certainly not advising organizations to even bother trying to come up with hard dollar figures for improved security; instead we suggest using costs as qualitative support for investment, rather than the main economic driver.

Packaging Your Business Case

The level of justification and packaging you need to get the project approved will depend on your specific environment. Some organizations need terribly formal documents, with spreadsheet models and ROI calculations. Others will accept a quick slide deck with the highlights, and pointers to more in-depth of analysis (to make sure the detailed analysis actually happened). Obviously do the least you need to get the green light for the project. The more time you spend concocting ROI models, the less time you have for fighting the bad guys.

Data Collection

Now we move on to how to select the right product/solution/service for your organization, and that involves digging into the technology. We start with the foundation of every SIEM and Log Management platform: data collection. This is where we gather data from the dozens of different types of devices and applications we monitor. 'Data' has a pretty broad meaning — here it typically refers to event and log records but can also include flow records, configuration data, SQL queries, and any other type of structured or non-structured data we want to pump into the platform for analysis.

It may sound simple, but being able to gather data from every hardware and software vendor on the planet in a scalable and reliable fashion is incredibly difficult. There is no single 'best' way to collect data, only those that fit with your business operations. With over 20 vendors in the Log Management and SIEM space, each using different terms to differentiate their products, it gets very confusing. In this series we will define vendor-neutral terms to describe the technical underpinnings and components of data collection, to level-set what you really need to worry about. In fact, while log files are what is commonly collected, we will use the term "data collection", as we recommend gathering more than just log files.

Data Collection Overview

Conceptually, data collection is very simple: we just gather the events from different devices and applications on our network to understand what is going on. Each device generates an event each time something happens, and collects them into a single repository known as a log file (although it might actually be a database). There are only four components to discuss for data collection, and each provides a pretty straightforward function. Here are the functional components:

- **Source** — There are many different sources — including applications, operating systems, firewalls, routers & switches, intrusion detection systems, access control software, and virtual machines — that generate data. We can even collect network traffic, either directly from the network or from routers that support NetFlow-style feeds.

- **Data** — This is the artifact telling us what actually happened. It could be an event, which is nothing more than a finite number of data elements to describe what happened. For example this might record someone logging into the system or a service failure. Minimum event data includes the network address, port number, device/host name, service type, operation being performed, result of the operation (success or error code), user who performed the operation, and timestamp. Or the data might just be configuration information or device status. In practice, event logs are pretty consistent across different sources — they all provide this basic information. But each offers additional data, including context. Additional data types may include things such as NetFlow records and configuration files. In practice, most of the data gathered will be events and logs, but we don't want to arbitrarily restrict our scope.

- **Collector** — This connects to a source device, directly or indirectly, to collect the events. Collectors take different forms: they can be agents residing on the source device (Fig. 1), remote code communicating over the network directly with the device (Fig. 2), or the source writing to either a log file, connector appliance or directly to a dedicated log

repository (Fig. 3). A collector may be provided by the SIEM vendor or a third party (normally the vendor of the device being monitored). Further, the collector functions differently depending upon the characteristics of the device. In most cases the source need only be configured once, and events will be pushed directly to the collector or into an external log file read by it. In some cases a collector must continually request data, polling the source at regular intervals.

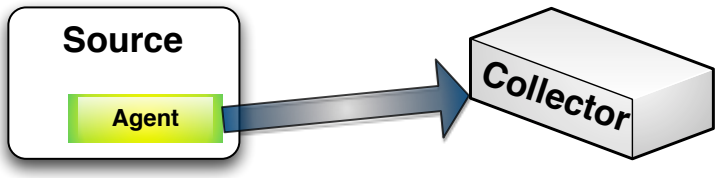


Fig 1. Agent data collector

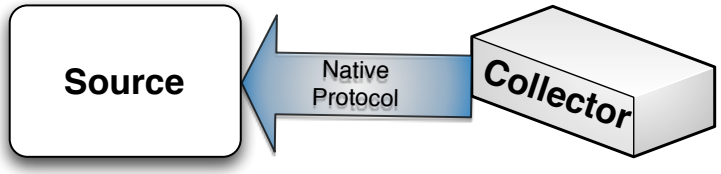


Fig 2. Direct connections to the device

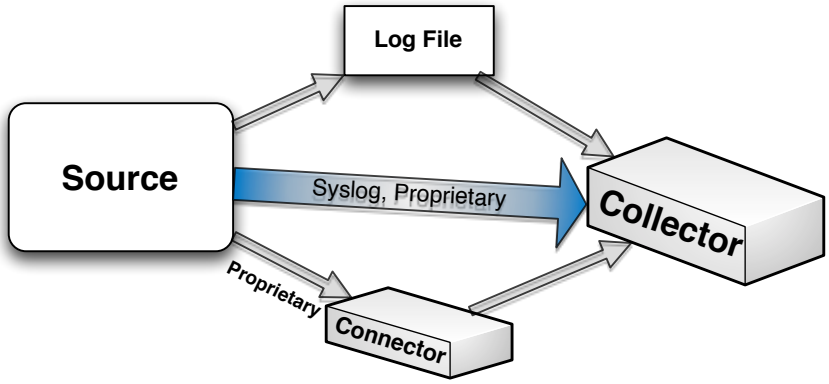


Fig 3. Log file collection

- **Protocol** — This is how collector communicates with a source. This is an oversimplification, of course, but think of it as a language or dialect the two agree on for communicating events. Unfortunately there are many of them! Sometimes the collector uses an API to communicate directly with the source (*e.g.,* OPSEC LEA APIs, MS WMI, RPC,

or SDEE). Sometimes events are streamed over networking protocols such as SNMP, NetFlow, or IPFIX. Sometimes the source drops events into a common file/record format, such as syslog, Windows Event Log, or syslog-ng, which is then read by the collector. Additionally, third party applications such as Lasso and Snare provide these features as a service.

Data collection is conceptually simple, but the thousands of potential variations make implementation a complex mess. It resembles a United Nations meeting: you have a whole bunch of people talking in different languages, each with a particular agenda of items they feel are important, and different ways they want to communicate information. Some are loquacious and won't shut up, while others need to be poked and prodded just to extract basic information. In a nutshell, it's up to the SIEM and Log Management platforms to act as the interpreters, gathering the information and putting it into some useful form.

Tradeoffs

Each model for data collection has trade-offs. Agents can be powerful proxies, allowing the SIEM platform to use robust (sometimes proprietary) connection protocols to safely and reliably move information off devices — in this scenario device setup and configuration are handled during agent installation. Agents can also take full advantage of native device features, and can tune and filter the event stream. But they have fallen out of favor since SIEM installations cover thousands of devices that creates a maintenance nightmare, requiring considerable time to install and maintain agents. Further, agents' processing and data storage requirements on the device can affect stability and performance. Finally, most agents require administrative access, which creates an additional security exposure on each device.

Another common technique streams events to log files, such as syslog or the Windows Event Log. These may reside on the device, stream to another server, or be sent directly to the log management system. The benefit of this method is that data arrives already formatted using a common protocol and layout. Further, if the events are collected in a file, this removes concerns about synchronization issues and uncollected events lost prior to collection — both problems when working directly with some devices. Unfortunately general-purpose logging systems require some data normalization, which can result in lost detail.

Some older devices, especially dedicated control systems, simply do not offer full-featured logging, and require API-level integration to collect events. These specialized devices are much more difficult to work with, and require dedicated full-time connections to collect event trails, creating both a maintenance nightmare and a performance penalty on each device. In these cases you do not have a choice, but need a synchronous connection in order to capture events.

Understand that data collection is not an either/or proposition. Depending on the breadth of your monitoring efforts, you may need to use each technique on some subset of device types and applications. Go into the project with your eyes open, recognizing the different types of collection, and the nuances and complexity of each.

Aggregation, Normalization, and Enrichment

Once we have all this data collected we need to understand how to put it into a manageable form for analysis, reporting, and long-term storage for forensics.

Aggregation

SIEM platforms collect data from thousands of different sources because these events provide the data we need to analyze the health and security of our environments. In order to get a broad end-to-end view, we need to consolidate what we collect onto a single platform. Aggregation is the process of moving data and log files from disparate sources into a common repository. Collected data is placed into a homogenous data store — typically purpose-built flat file repositories or relational databases — where analysis, reporting, and forensics occur; archival policies are also applied here.

The process of aggregation — compiling these dissimilar event feeds into a common repository — is fundamental to Log Management and most SIEM platforms. Data aggregation can be performed by sending data directly into the SIEM/LM platform (which may be deployed in multiple tiers), or an intermediary host can collect log data from the source and periodically move it into the SIEM system. Aggregation is critical because we need to manage data in a consistent fashion: security, retention, and archive policies must be systematically applied. Perhaps most importantly, having all the data on a common platform allows for event correlation and data analysis, which are key to the uses we have described.

There are some downsides to aggregating data onto a common platform. The first is scale: analysis becomes exponentially harder as the data set grows. Centralized collection means huge data stores, greatly increasing the computational burden on the SIEM/LM platform. Technical architectures can help scale, but ultimately these systems require significant horsepower to handle an enterprise's data. Systems that utilize central filtering and retention policies require all data to be moved and stored — typically multiple times — increasing the burden on the network.

Some systems scale using distributed processing, where filtering and analysis occur outside the central repository, typically at the distributed data collection point. This reduces the computational burden on the central server and allows processing to occur on smaller, more manageable data sets. It does require that policies, along with the code to process them, be distributed and kept current throughout the network. Distributed agent processes are a handy way to "divide and conquer", but increase IT administration requirements. This strategy also adds a computational burden to the data collection points, degrading their performance and potentially slowing them enough to drop incoming data.

Data Normalization

If the process of aggregation is to merge dissimilar event feeds into one common platform, normalization takes it one step further by reducing the records down to just common event attributes. As we mentioned in the data collection section, most data sources collect exactly the same base event attributes: time, user, operation, network address, and so on. Facilities like syslog not only group the common attributes, but provide mechanisms to collect supplementary information that does not fit the basic template. Normalization is where known data attributes are fed into a generic template, and anything that doesn't fit is simply omitted from the normalized event log. After all, to analyze we want to compare apple to apples, so we throw away any oranges for the sake of simplicity.

Depending on the SIEM or Log Management vendor, the original non-normalized records may be kept in a separate repository for forensics purposes prior to later archival or deletion, or they might simply be discarded. In practice, discarding original data is a bad idea, because the full records are required for any kind of legal action. So most products keep the raw event logs for a user-specified period prior to archival. In some cases, the SIEM platform keeps a link to the original event in the normalized event log, which provides 'drill-down' capability to easily reference extra information collected from the device.

Normalization allows for predicable and consistent storage for all records, and indexes these records for fast searching and sorting — key when battling the clock to investigate an incident. Additionally normalization enables basic and consistent reporting and analysis on every event regardless of the data source. When the attributes are consistent, event correlation and analysis are much easier.

Enriching the Future

We are starting to see a number of platforms doing enrichment, adding supplemental information (such as geo-location, transaction numbers, application data, etc.) to logs and events to enhance analysis and reporting. Enabled by cheap storage and Moore's Law, and driven by ever-increasing demand to collect more information to support security and compliance efforts, we expect SIEM/LM platforms to increase enrichment capabilities. Data enrichment requires a highly scalable technical architecture, purpose-built for multi-factor analysis and scale, making tomorrow's SIEM/LM platforms look very similar to current business intelligence platforms.

But that just scratches the surface in terms of enrichment, because data from the analysis can also be added back to the records. Examples include identity matching across multiple services and devices, behavioral detection, transaction IDs, and even rudimentary content analysis. This is somewhat like having the system take notes and extrapolate additional meaning from the raw data, making the original record more complete and useful. This is a new concept for SIEM, so what enrichment will ultimately encompass is anyone's guess. But as the core functions of SIEM have standardized, we expect vendors to introduce new ways to derive additional value from the seas of data they collect.

Reporting and Forensics

We have pushed, prodded, and poked at the data to get it into a manageable format, now we need to put it to use. Reports and forensic analysis are the features most users work with on a day to day basis, since any operational function is going to be triggered by an alert or report and investigated using forensic information. Collection, normalization, correlation, and all the other things we do are just to get us to the point where we can conduct forensics and report on our findings. Ultimately, these features are critical to the success of your SIEM/LM deployment, so while we'll dig in to describe how the technology works, we will also discuss what to look for when making buying decisions.

Reporting

For those of us who have been in the industry for a long time, the term 'reporting' brings back bad memories. It evokes hundreds of pages of printouts on tractor feed paper, with thousands of entries, each row looking exactly the same. It brings to mind hours of scanning these lines, yellow highlighter in hand, marking unusual entries. It brings to mind the tailoring of reports to include new data, excluding unneeded columns, importing files into print services, and hoping nothing got messed up which might require restarting from the beginning.

Those days are fortunately long gone, as SIEM and Log Management have evolved their capabilities to automate a lot of this work, providing graphical representations that allow viewing data in novel ways. Reporting is a key capability because this process was just plain hard work. To evaluate reporting features included in SIEM/LM, we need to understand what it is, and the stages of a reporting process. The term 'reporting' is a colloquialism used to encompass a group of activities: selecting, formatting, moving, and reviewing data are all parts of the reporting process.

At its simplest reporting is just selecting a subset of the data we previously captured for review, focused analysis, or a permanent record ('artifact') of activity. Its primary use is to put data into an understandable form, so we can analyze activity and substantiate controls without having to comb through lots of irrelevant stuff. The report comprises the simplified view needed to facilitate review or, as we will discuss later, forensic analysis. We also should not be constrained by the traditional definition of a report, which is a stack of papers (or in modern days a PDF). Our definition of reporting can embrace views within an interface that facilitates analysis and investigation.

The second common use is to capture and record events that demonstrates completion of an assigned task. These reports are historic records kept for verification. Trouble-ticket work orders and regulatory reports are common examples, where a report is created and 'signed' by both the producer of the report and an auditor. These snapshots of events may be kept within, or stored separately from, the SIEM/LM system.

There are a couple basic aspects to reporting that we pay close attention to when evaluating SIEM/LM reporting capabilities:

- What reports are included with the standard product?

- How easy is it to manage and automate reports?
- How easy is it to create new, ad-hoc reports?
- What export and integration options are available?

For many standard tasks and compliance needs, pre-built reports are provided by the vendor to lower costs and speed up product deployment. At minimum, vendors provide canned reports for PCI, Sarbanes-Oxley, and HIPAA. We know that compliance is the reason many of you are reading this paper, and will be the reason you invest in SIEM/LM. Reports embody the tangible benefit to auditors, operations, and security staff. Just keep in mind that 2000 built-in reports is not necessarily better than 100, despite vendor claims. Most end users typically use 10-15 reports on an ongoing basis, and the production of those customized reports must be automated based on the user's requirements.

Most end users want to feel unique, so they like to customize the reports — even if the built-in reports are fine. But there is a real need for ad-hoc reports in forensic analysis and implementation of new rules. Most policies take time to refine, making sure only the needed data is collected, and ensuring completeness and accuracy. So the reporting engine needs to make this process easy, or the user experience suffers dramatically.

Finally, the data within the reports is often shared across different audiences and applications. The ability to export raw data for use with third-party reporting and analysis tools is important, and demands careful consideration during selection.

The value of these systems largely resides within their interface and reports. We call them collectively user experience, and although many security professionals minimize the focus on reporting during the evaluation process that can be a critical mistake. Reports are how you will show value from the SIEM/LM platform, so make sure the engine can support the information you need to show in the way you need to show it.

Forensics

Forensic analysis is like black magic. There are some provocateurs in the security research space who believe you can largely automate forensics with a set of craftily created monitoring policies. From where we sit, that's hogwash. If we knew in advance what to look for, there would be no reason to wait until afterward to perform the analysis — instead we would alert on it. And this is really the difference between *alerting* and *forensic analysis*. We need to correlate data from multiple sources and have a real live human being make a judgement call about the severity and urgency of any potential incident. Let's be clear: these pseudo-analyst claims and vendor promotional fluff are complete BS, and do a disservice to end users by creating absurd expectations.

Let's take a step back. Forensic analysis is conducted by skilled security and network analysts to investigate an event, or more likely a sequence of events, to isolate indications of fraud or misuse. An analyst may have an idea what to look for in advance, but more often you don't actually know what you are looking for and need to navigate through thousands of events to piece together what happened and understand the breadth of the damage. This involves rewriting queries over and over to drill down and look at data, using different methods of graphing and visualization before finding the proverbial needle in the haystack.

The uses for forensic analysis are numerous, including examination of past events and data to determine what happened in your network, OS, or application. This may be to verify something that was supposed to happen actually occurred, or

to better understand whether strange activity was fraud or misuse. You might need forensic analysis to perform simple health checks on equipment and business operations. You could need forensics to monitor user activity to support disciplinary actions against employees. You might even need to provide data to law enforcement to pursue criminal prosecution in a data breach scenario.

Unlike correlation and alerting, where analysis of events is largely automated, forensic analysis is largely manual. Fortunately we can leverage collection, normalization, and correlation activities — much of the data has already been collected, aggregated, and indexed within the SIEM/LM platform.

A forensic analysis usually starts with data provided by a report, an alert, or a query against the SIEM/LM repository. We start with an idea of whether we are interested in specific application traffic, strange behavior from a host, or pretty much an infinite number of things that might be suspicious. We select data with the attributes we are interested in, gathering needed information to analyze events and validate whether the initial suspicious activity is much ado about nothing, or indicates a major issue.

These queries might be as simple as "Show all failed logins for user 'mrothman'", or as specific as "Show events from all firewalls, between 1 and 4 am, that involved this list of users." It is increasingly common to examine application-layer or database activity to provide context for business transactions — for example, "list all changes to the general ledger table where the user was not 'GA_Admin' or the application was not 'GA_Registered_App'.

We need a couple key capabilities to effectively perform forensic analysis:

- Custom queries and views of data in the repository
- Access to correlated and normalized data
- Drill-down to view non-normalized or supplementary data
- Ability to reference and access older data
- Speed, since forensics is usually a race against time (and attackers)

Basically the most important capability is to enable a skilled analyst to follow their instincts. Forensics is all about making their job easier by facilitating access, correlation, and viewing of data. They may start with a set of anomalous communications between two devices, but end up looking at application logs and database transactions to prove a significant data breach. If queries take too long, data is manipulated or discarded, or data is not collected, the investigator's ability to do his/her job is hindered. So the main role of SIEM/LM in forensics is to streamline the investigative process.

To be clear, the tool only makes the process faster and more accurate. Without a strong incident response process, no tool can solve the problem. Although we all get very impressed by a zillion built-in reports and cool drill-down investigations during a vendor demo, don't miss the forest for the trees. SIEM/Log Management platforms can only streamline a process that already exists. And if the process is bad, you'll just execute on that bad process faster.

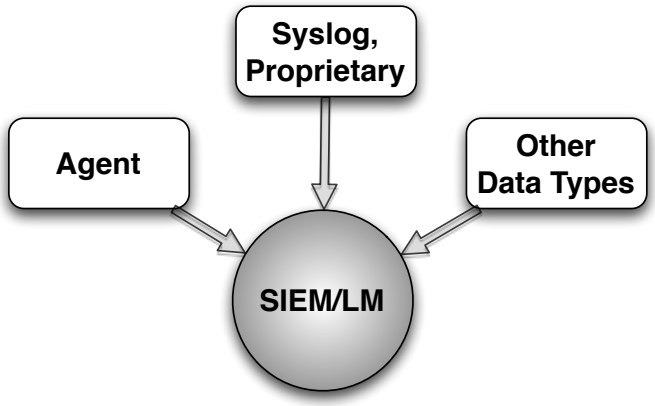
Deployment Models

We have covered the major features and capabilities of SIEM and Log Management tools, so now let's discuss architecture and deployment models. Each architecture addresses a specific issue, such as (relative) simplicity, coverage for remote devices, scaling across hundreds of thousands of devices, real-time analysis, or handling millions of events per second. Each has advantages and disadvantages in complexity, analysis performance, reporting performance, scalability, storage, and cost.

There are four models to discuss: 'flat' central collection, hierarchical, ring, and mesh. And none of these models is mutually exclusive. Some regions may deploy a flat model, but send information up to a central location through a hierarchy. These are not absolutes — just guidelines to consider as you design your deployment to address your project drivers.

Flat

The original deployment model for SIEM and Log Management platforms was a single server that collected and consolidated log files. In this model all log storage, normalization, and correlation occurs within a central appliance. All data collection methods (agent, flow, syslog, etc.) are available, but data is always stored in the same central location.

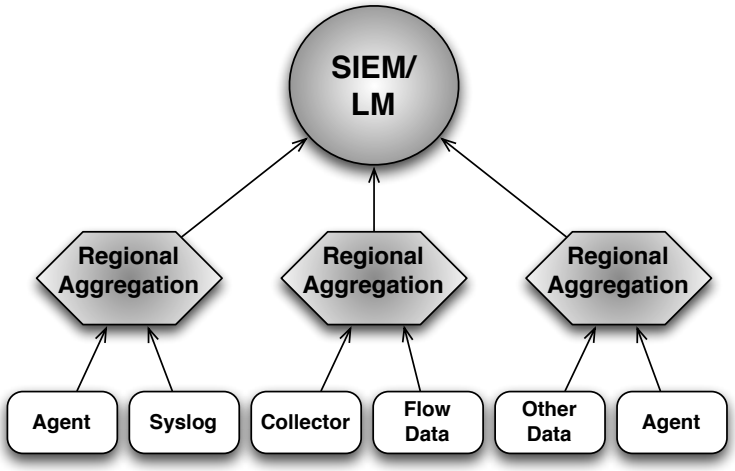


A flat model is far simpler to deploy. All data and policies are consolidated, so there are no policy or data synchronization issues. But of course ultimately a flat central collection model is limited in scalability, processing, and the quantity of data it can manage. A single installation provides a fixed amount of processing and storage, and reporting becomes progressively harder and slower as data sets grow. Truth be told, we only see this kind of architecture for "checkbox compliance", predominately for smaller companies with modest data collection needs.

The remaining models address the limitations of this base architecture.

Hierarchical

The hierarchical model consists of a central SIEM server, similar to the flat model above. Rather than communicating directly with endpoints where data is collected, the central SIEM server acts as a parent, and communicates with intermediary collection appliances (children). Each child collects data from a subset of devices, typically from a specific region or location. The child nodes collect and store data, then normalize events before passing them along to the central SIEM server for aggregation, correlation, and reporting. Raw event data remains on the local children for forensic purposes.

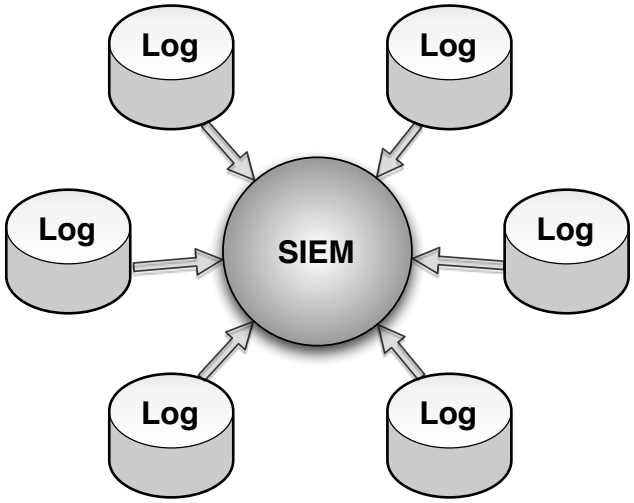


The hierarchical model was introduced to help scale across larger organizations, where it wasn't practical to send the raw data streams across the network and some level of storage tiers were required. The hierarchical model helps divide and conquer data management challenges by distributing load across a larger number of engines, and reduces network overhead by only passing a subset of the captured data to the parent for correlation and analysis. Data storage, backup, and processing are much easier on smaller data sets. Further, construction of reports can be distributed across multiple nodes — important for very large data sets.

There are many variations on this model, but the primary point is that the parent and child nodes take on different responsibilities. Alerting, filtering, normalization, reporting, and anything else having to do with policy enforcement can be part of the parent or the child, but not both. The good news is you can scale up by adding new child nodes. The downside is that every function handled by the child nodes requires synchronization with the server. For example, alerting is faster from the child node, but requires distribution of the code and policies. Further, alerting from the child node(s) lacks correlation of events happening elsewhere in the environment to refine their accuracy. Despite the trade-offs, this hierarchical model is very flexible.

Ring

In the Ring model — otherwise known as the Moat — you have a central SIEM server ringed by many log collection devices. Each logger in the ring is responsible for collecting data from event sources. These log archives are also used to support distributed reporting. The log devices send a normalized and filtered (substantially reduced) stream of events to the master SIEM device. The SIEM server sitting in the middle is responsible for correlation of events and analysis. This architecture was largely designed to address scalability limitations with some SIEM offerings (typically built on relational databases). It wasn't cost effective to scale the SIEM engine to handle mushrooming event traffic, so surrounding the SIEM centerpiece with logging devices allowed it to analyze the most critical events while providing a more cost-effective scaling mechanism.

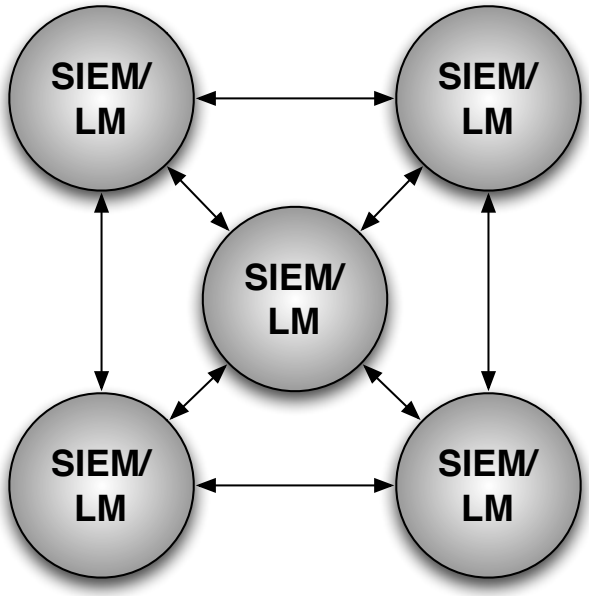


The upside of this model is that simple (cheaper) high-performance loggers do the bulk of the heavy lifting, and the expensive SIEM components perform the meat of the analysis. This model addresses scalability and data management issues, while reducing the need to distribute code and policies among different devices.

There are a couple of issues with the ring model. The biggest problem remains a lack of integration between the two systems. Management tools for the data loggers and the SIEM may be linked together with some type of dashboard, but you quickly discover the two-headed monster of two totally separate products under the covers. Similarly, log management vendors were trying to graft better analysis and correlation onto their existing products, resulting in a series of acquisitions that provided log management players with SIEM. Either way, you end up with two separate products trying to solve a single problem. This is not a happy "you got your chocolate in my peanut butter" moment, and will continue to be a thorny issue for customers until vendors fully integrate their SIEM and Log Management offerings, instead of slapping on dashboard band-aids and claiming the products are fully integrated.

Mesh

The last model we want to discuss is the mesh deployment. The mesh is a group of interrelated systems, each performing full log management and SIEM functions for part of the environment. Basically this is a cluster of SIEM/LM appliances; each a *functional peer* with full analysis, correlation, filtering, storage, and reporting for local events. The servers can all be linked together to form a mesh or deployed hierarchically, depending on customer needs.



While this model is more complex to deploy and administer, and requires a purpose-built data store to manage high-speed storage and analysis, it does solve several problems. For organizations that require segregation of both data and duties, the mesh model is unmatched. It provides the ability to aggregate and correlate specific segments or applications on specific subsets of servers deployed organizationally or geographically, making analysis and reporting flexible. Unlike the other models, it can divide and conquer processing and storage requirements flexibly depending on the requirements of the business, rather than the scalability limitations of the product being deployed.

Each vendor's platform is capable implementing two or more of these models, but generally not all of them. Each product's technical design (particularly the datastore) dictates which deployment models are possible. Additionally, the level of integration between the SIEM and Log Management components has an effect as well. As we said in our introduction, every SIEM vendor offers some degree of log management capability, and most Log Management vendors offer SIEM functions. This does not mean that the offerings are fully integrated by any stretch of the imagination. Deployment and management costs are clearly affected by product integration or lack thereof, so make sure to perform due diligence in the purchase process to understand the underlying product architecture, and the limitations and compromises necessary to make the product work in your environment.

Data Management

We covered SIEM and Log Management deployment architectures in depth to underscore how different models are used to deal with scalability and data management issues. In some cases these deployment choices are driven by the underlying data handling mechanism within the product. Each platform stores and manages data differently, which has a significant impact on product scalability, data management, and reporting & forensics capabilities. Here we discuss the different data storage models, with their advantages and disadvantages.

Relational Database

In the early days of this technology, most SIEM and log management systems were built on relational database engines to store events and log records. In this model the SIEM platform maps data attributes from each data source into database columns, so each event is stored in a single database row. There are numerous advantages to this model, including:

- **Data Validation** — As data is inserted into the column, the database verifies data type and range. Integrity check failures indicate corrupted files and are omitted from the import, with notification to administrators.
- **Event Consistency** — An event from a Cisco router now looks just like an event from a Juniper router, and *vice-versa*, as events are normalized before being stored in the table.
- **Reporting** — Reports are easier to generate from validated data columns, and the database can format data when generating the report. Reports run far faster thanks to column indices, effectively filtering and ordering events.
- **Analytics** — An RDBMS facilitates complex queries across all available attributes, inspected content, and correlation.

This model for data storage has fallen out of favor due to the overhead of data insertion: as each row is inserted the database must perform the checks and periodically rebuild indices. As daily volumes scaled from millions of events to hundreds of millions and billions, this overhead became problematic and resulted in significant scalability issues with SIEM offerings built on RDBMS.

Further, data that does not fit into the tables defined in the relational model is typically left out. Unless there is some other method for maintaining the fidelity and integrity of the original event records, this is problematic for forensics. This "selective memory" can also result in data accuracy issues, as truncated records may not correlate properly and can hamper analysis.

As a result SIEM/LM architectures based on RDBMS are waning, as products in this space re-architect their backend data stores to address these issues. On the other hand, RDBMS storage is not totally dead — some vendors have instead chosen to streamline data insertion, basically by turning off some RDBMS checks and integrity verification. Others use an RDBMS to supplement a flat file architecture (described below), leveraging the advantages above for reporting and forensics.

Flat File

Flat files, or just 'files', are now the most common way to store events for SIEM and Log Management. Files serve as a blank canvas for the vendor; as they can introduce any structure they choose to help define, format, and delineate events. Anything that helps with correlation and speeds up future searches is included, and each vendor has their own secret sauce for building files. Each file typically contains a day's events, possibly from a single source, with each event clearly delineated. The files (and in some cases each event) can be tagged with additional information — this is called "log enrichment". These tags offer some of the contextual benefits of a relational database and help define attributes. Some even include a control structure similar to VSAM files. The events may be stored in their raw form or normalized prior to insertion. Flat files offer several advantages.

- **Performance** — Since normalization (to the degree necessary) happens before data insertion, there is very little work to be performed prior to insertion compared to a relational database. Data is stored as quickly as the physical media can handle, and often available immediately for searching and analysis.
- **Flexibility** — Stored events are not limited to specific normalized columns as they are in a relational database, but can take any form. Changes to internal file formats are much easier.
- **Search** — Searches can be performed without understanding the underlying structures, using simple keyword search. At least one log management vendor provides a Google-style search capability across data files. Alternately, search can rely on tags and keywords established by the vendor.

The flat file tradeoffs are twofold. First, any data management capabilities — such as indexing and data integrity — must be built from scratch by the vendor, since no RDBMS features are provided by the underlying platform. This means the SIEM/LM vendor must provide any needed facilities for data integrity, normalization, filtering, and indexing. Second, there is an efficiency tradeoff. Some vendors tag, index, and normalize prior to insertion; others initially record raw events, later re-reading the data in order to normalize it, and then rewrite the reformatted data. The later method offers faster insertion at the expense of greater total storage and processing requirements.

The good news is that a few years ago most vendors saw the scalability wall of RDBMS approaching, and began investing in their own back-end data management environments. At this point many platforms feature purpose-built high-performance data stores, and we believe this will be the underlying architecture for these products moving forward.

Advanced Features

We've already discussed the basic features of a SIEM/Log Management platform, including collection, aggregation and normalization, correlation and alerting, reporting and forensics, and deployment architectures. But these are the core functions, and part of what each product brings to the table.

As markets evolve and vendors push to differentiate themselves, more and more capabilities get integrated into the platforms. In the case of SIEM/LM, this means pumping more data into the analysis engine, and making the engines themselves smarter. The idea is to make 1+1 produce 5, as multiple data types provide more insight than a single source — that's the concept, anyway. To be clear, having more data does not make directly the product any better. The only way to really leverage additional data is to build correlation rules, alerts, and reports that utilize the extra data.

Let's take a tour through some of the advanced data types you'll see integrated into SIEM/LM platforms.

Flow

Network flows are the connection records that stream out of a router or switch. These small and simple data files/streams typically list source, destination, and packet type. Flow data was really the first new data type which, when integrated with event and log data, actually made SIEM/LM smarter. Flow data allowed the system to establish a baseline and scan for anomalous network traffic as the first indication of a problem.

An entire sub-market of network management — network behavioral analysis — revolves around analyzing and visualizing flow data to understand the traffic dynamics of networks, and pinpointing performance and capacity issues before they impact users. Many of the NBA vendors have been unsuccessfully trying to position their products in the security market, but when combined with events and logs, flow data is very useful.

As an example, consider a typical attack where a web server is compromised and then used as a beachhead to further compromise an application server and the backend database server. The data needs to be exfiltrated in some way, so the attackers establish a secure pipe to an external zombie device. But the application server doesn't typically send data to external devices, so flow data would show an anomalous traffic flow. At that point an administrator could analyze the logs, with correlated activity showing a new account created on the database server, and identify the breach.

Could an accurate correlation rule have caught the reconnaissance and subsequent compromise of the servers? Maybe. But the network doesn't lie, and at some point the attackers need to move the data out. These types of strange network flows can be a strong indication of a successful attack, but remember strange flows only appear after the attack has occurred. So flow data is really for reacting faster to attacks already underway.

Even more powerful is the ability to set up compound correlation rules, which factor in specific events and flow scenarios. Of course setting up these rules is complicated and they require a lot of tuning, but once the additional data stream is in place, there are many options for leveraging it.

Identity

Everyone wants to feel like more than just a number, but when talking about SIEM/Log Management, your IP address is pretty much who you are. You can detect many problems by just analyzing all traffic indiscriminately, but this tends to generate plenty of false positives. What about the scenario where the privileged user makes a change on a critical server? Maybe they used a different device, which had a different IP address. This would show up as unusual for that action and could trigger an alert.

But if the system were able to leverage identity information to know the same privileged user was making the change, all would be well, right? That's the idea behind identity integration with SIEM/LM. Basically, the analysis engine pulls in directory information from the major directory stores (Active Directory & LDAP) to understand who is in the environment and what groups they belong to, which indicates what access rights they have. Other identity data — such as provisioning and authentication information — can be pulled in to enable advanced analysis, such as identifying a terminated user accessing a key system.

The holy grail of identity integration is user activity monitoring. Yup, Big Brother lives — and he always knows exactly what you are doing. In this scenario you'd be able to set up a baseline for a group of users (such as Accounting Clerks), including which systems they access, who they communicate with, and what they do. There are actually a handful of other attributes that help identify a single user even when using generic service accounts. Then you can look for anomalies, such as an accounting clerk accessing the HR system, making a change on a sensitive server, or even sending data to his/her Gmail account. This isn't a smoking gun, *per se*, but it does give administrators a place to look for issues.

Again, additional data types beyond plain event logs can effectively make the system smarter and streamline problem identification.

Database Activity Monitoring

Recently SIEM/LM platforms have been integrating Database Activity Monitoring (DAM) data, which collects very detailed information about what is happening to critical data stores. As with the flow data discussed above, DAM can serve up activity and audit data for SIEM. These sources not only provide more data, but add context, helping with both correlation and forensic analysis. Securosis has published plenty of information on DAM, which you can check out in our research library <http://securosis.com/research/publication/report-selecting-a-database-activity-monitoring-solution/>.

The purpose of DAM integration is to drive analysis deeper into database transactions, gaining the ability to detect patterns which indicate successful compromise or misuse. As a simple example, if a mobile user gets infected at Starbucks (like *that* ever happens!) and then unwittingly provides access to the corporate network, the attacker then proceeds to compromise the database.

The DAM device monitors the transactions to and from the database, and should see the attack traffic. At that point the admin must go to another system to figure out the issue with the rogue device. But if all the data is available within a single platform, the admin would be able to instantly know the database breach resulted from the compromised device and remediate.

Additionally, powerful correlation rules can be set up to look for account changes and other significant events on certain servers, followed up by a data dump from the database (recorded by DAM), and a bulk file transfer to an external location

(detectable in flow data). This is certainly getting closer to a smoking gun — even better if the attack scenarios are modeled and implemented as rules in the SIEM/LM.

Application Monitoring

Like DAM, some SIEM/LM platforms are climbing up to the application layer by ingesting application logs, as well as performing simple content analysis on specific application types — typically email or web traffic. Again, baseline models can identify how applications should behave; then alerts can be set up for behavior which is *not normal*.

The problem with application monitoring is the amount of work required to get it right. Each application works differently in each environment, so significant tuning is required to get the rules right and tighten thresholds enough to provide relevant alerts. With the number of applications in a typical organization, getting useful coverage within any environment takes substantial time and resources.

But over time this capability will continue to improve and become much more useful in practice. We expect that to take another 12-18 months.

Configuration Monitoring

Another data type useful for SIEM/LM integration is configuration data. This means different things to different folks, so let's level set a bit. We are referring to the configuration settings for security, applications, network, and computing devices. Most attacks involve some type of configuration change, whether it's a service turned on or off, a new user added, a new protocol allowed, or a destination address added. Monitoring these configurations can provide early detection of attacks in progress.

This integration can happen with the SIEM/LM platform directly collecting and monitoring the configurations (many of the vendors can monitor network and security device configurations) or by taking an alert or log stream from a standalone configuration monitoring product. Either works, so long as the correlation rules and reports within the SIEM are built to take advantage of the configuration data.

Let's run through a quick scenario. In the event of an attack on a retailer's Point of Sale (POS) system, events would show reconnaissance activity on a store wireless network, which happens frequently and wouldn't trigger any alarms. The attacker then breaks the WEP key and gains access to the POS network, ultimately compromising the POS devices, which run an un-patched version of embedded Windows XP. Yes, I know this organization deserves to be 'pwned' for using WEP and un-patched XP, but indulge us a bit. None of this would necessarily be caught through typical event logs.

Then the attacker enables FTP on the POS terminal, which would change the configuration and be detected by configuration monitoring. So the admin can investigate FTP being active on a POS device, which indicates badness. Combine that with other event and logging activity, and a case for further investigation can be made — which is the point of having the SIEM/LM platform in the first place.

File Integrity Monitoring

The last data type we'll discuss is file integrity data. This involves monitoring for changes to key system files such as the IP stack. If one of these files changes and it's not traceable to legitimate patching, it usually indicates some kind of unauthorized activity. So this data is similarly useful for helping to narrow the scope of analysis for a security analyst.

If the analyst sees system files changing on a critical server, in combination with strange network flows, other configuration changes, and IDS alerts, that's a good indication of a successful attack. Remember, it isn't necessary to find a smoking gun. SIEM/LM is useful if it can make us a bit smarter and enable a security analyst to react faster to an attack by helping to focus on where to look and what to investigate.

Direct or Indirect Integration?

One of the ways vendors try to differentiate is whether their product takes the data in directly and does the analysis within the SIEM/LM platform, or partners with leading vendors of standalone solutions — such as NBA and configuration monitoring. We aren't religious one way or the other.

There are advantages to direct integration — all the data is in one location, which facilitates forensic investigation; this may also enable more detailed correlation rules and compliance reports. On the other hand, a standalone NBA system is more useful to the network administrator, at the expense of fewer capabilities built into the SIEM. If it's the network administrator's budget they will buy NBA, and the security team will get alerts. Either way is fine, since it's about making the SIEM/LM smarter and focusing investigations.

Additional Data = More Complexity

As we described in our introduction, making SIEM/LM work is a fairly complicated endeavor, and that's just dealing with logs and events. When you add a couple or more additional data types you multiply the number of rules and reports the system can generate. Couple that with enrichment and activity profiles, and you have seriously increased complexity. That can be useful by supporting broader analysis, and also painful because tuning and performance become bigger issues, so be careful what you wish for.

Ultimately, the use cases need to drive the advanced features evaluated during the procurement process. If you are just trying to meet a compliance automation requirement, then flow data may not be that useful and shouldn't weigh heavily in the selection process. But if you are trying to gain operational efficiencies, then something like configuration monitoring should allow your analysts to kill two birds with one platform, so the supported data types become more important.

Integration

They say that no man is an island, and in the security space that's very true. No system is, either — especially those tasked with some kind of security management. We get caught up in SIEM and Log Management platforms' ability to suck in every piece of information possible to help with event correlation and analysis, but when it comes down to it security management is just one aspect of an enterprise's management stack. SIEM/Log Management is only one discipline in the security management chain, and must feed some portion of its analysis to supporting systems. So clearly integration is key, both to getting value from SIEM/LM, and to making sure the rest of the organization is on board with buying and deploying the technology.

For a number of enterprise IT management systems it is important to integrate with the SIEM/Log Management platform, ranging from importing data sources, to sending alerts, and even participating in an IT organization's workflow. We have broken the integrations into inbound (receiving data from another tool) and outbound (sending data/alerts to another tool).

Inbound integration

- **Security management tools** — We discussed this a bit when talking about data collection, regarding the importance of broadening the number of data sources for analysis and reporting. These systems include vulnerability management, configuration management, change detection, network behavioral analysis, file integrity monitoring, endpoint security consoles, etc. Typically integration with these systems occurs via custom connectors, and most SIEM/LM players have relationships with the big vendors in each space.

- **Identity Management** — Identity integration was discussed as an advanced feature and is another key input to the SIEM/LM platform. This can include user and group information (to streamline deployment and ongoing user management) from enterprise directory systems like Active Directory and LDAP, as well as provisioning and entitlement information to implement user activity monitoring. These integrations tend to be via custom connectors as well.

Because these inbound integrations tend to require custom connectors to get proper breadth and fidelity of data, it's a good idea to learn a bit about each vendor's partner program. Vendors use these programs to gain access to the engineering teams behind their data sources; but also pay attention as their vehicles for developing rules, policies, and reports to take advantage of the additional data.

Outbound integration

- **IT GRC** — Given that SIEM/Log Management gathers information useful to substantiate security controls for compliance purposes, clearly it would be helpful to send that information to a broader IT GRC (Governance, Risk, and Compliance) platform presumably managing the compliance process at a higher level. So integration with whatever IT GRC platform is in use within your organization (if any) is an important consideration for deciding to acquire SIEM/Log Management technology.

- **Help Desk** — The analysis performed within the SIEM/Log Management platform provides information about attacks in progress and usually requires some type of remediation action once an alert is validated. To streamline fixing these issues, it's useful to be able to submit trouble tickets directly into the organization's help desk system to close the loop. Some SIEM/Log Management platforms include built-in trouble ticket systems, but we've found that capability is infrequently used because companies large enough to utilize SIEM/LM already have some kind of help desk system. Look for the ability to not only send alerts (with sufficient detail to allow the operations team to quickly fix the issue), but also to receive information back when a ticket is closed, and to automatically close the alert within the SIEM platform.
- **CMDB** — Many enterprises have also embraced configuration management database (CMDB) technology to track IT assets and ensure that configurations adhere to corporate policies. When trying to ensure changes are authorized, it's helpful to be able to send indications of changes at the system and/or device level to the CMDB for confirmation.

Again, paying attention to each vendor's partner program and announced relationships can yield valuable information about the frequency of true enterprise deployment, as large customers demand their vendors work together — often forcing some kind of integration. It also pays to ask vendor references about their integration offerings, because issuing a press release does not mean the integration is functional, complete, or useful. As with everything else in the security industry, don't believe everything you read. Trust, but verify.

Selection Process

Now that you thoroughly understand the use cases and technology underpinning SIEM and Log Management platforms, it's time to flex your knowledge and actually buy one. As with most of our research at Securosis, we prefer to map out a very detailed process, and leaving you to decide which steps make sense in your situation. We don't expect every organization to go through every step in this process —figure out what will work for your organization and do that.

Define Needs

Before you start looking at any tools, you need to understand why you might need a SIEM/LM; how you plan on using it; and the business processes around management, policy creation, and incident handling. You can (and should) consult our descriptions of the use cases to really understand what problem you are trying to solve and why. If you don't understand this your project is doomed to fail. And that's all we'll say about that.

- **Create a selection committee** — Yes, we hate the term 'committee' as well, but the reality is that a decision to acquire SIEM — along with the business issues it is expected to address — comes from multiple groups. SIEM/LM touches not only the security team, but also any risk management, audit, compliance, and operational teams as well. So it's best to get someone from each of these teams (to the degree they exist in your organization) on the committee. You want to ensure that anyone who could say no, or subvert the selection at the 11th hour, is on board from the beginning. Sorry, that involves playing the political game, but if you want to get the process over the finish line, you'll do what you need to.

- **Define the systems and platforms to monitor** — Are you looking to monitor just security devices or also general-purpose network equipment, databases, applications, VMs, and/or anything else? In this stage, detail the monitoring scope and the technical specifics of the platforms involved. You'll use this list to determine technical requirements and prioritize features and platform support later in the selection process. Remember that your needs will grow over time and you may be limited by budget during the initial procurement, so break the list into a group of high priority sources with immediate needs, and other groups of other data sources you may want to monitor later.

- **Determine security and/or compliance requirements** — The committee really helps with collecting requirements, as well as mapping out reports and alerts. The implementation will involve some level of correlation, analysis, reporting, and integration— which needs to be defined ahead of time. Obviously that can and will change over time, but give this some thought because these requirements will drive your selection. You don't need to buy a Rolls-Royce if a Nissan Sentra would solve your requirements. In this step map your security and compliance needs to the platforms and systems from the previous step, which helps determine everything from technical requirements to process workflow.

- **Outline process workflow, forensics, and reporting requirements** — SIEM/LM workflow is highly dependent on usage. When used in a security context, the security team monitors and manages events, and will have an escalation process to verify attacks and remediate. When used to improve efficiency, the key is to leverage as many rules and alerts as possible, which is really a security team function. Forensics use will involve the investigative/incident team. In

most cases audit, legal, and/or compliance will have at least some sort of reporting role, because compliance is typically the funding source for the project. Different SIEM/LM platforms have different strengths and weaknesses in terms of management interfaces, reporting, forensics, and internal workflow, so knowing your process before defining technical requirements can prevent headaches down the road.

- **Product versus managed service** — Are you open to using a managed service for SIEM/LM? Do you have the internal resources/expertise to manage (and tune) the platform? Now is the time to decide whether a service is an option, as that impacts the rest of the selection process.

By the end of this phase you should have defined key stakeholders, convened a selection team, prioritized the systems to protect, determined protection requirements, and roughed out workflow needs.

Formalize Requirements

This phase can be performed by a smaller team working under the mandate of the selection committee. Here the generic needs determined in phase 1 are translated into specific technical features, and any additional requirements are considered. This is the time to come up with criteria for collection and aggregation, additional infrastructure integration, data storage/archival, deployment architecture, management and identity integration, and so on. You may need to dig into what information your devices provide to ensure you can collect the necessary data to reliably feed the SIEM platform. You can always refine these requirements as you proceed through the selection process and get a better feel for how the products work.

At the conclusion of this stage you develop a formal RFI (Request For Information) to release to vendors, and a rough RFP (Request For Proposals) that you'll clean up and formally issue in the evaluation phase.

Evaluate Products

All the SIEM/LM vendors tell similar stories, which makes it difficult to cut through the marketing and figure out whether a product really meets your needs. The following steps should minimize your risk and help you feel confident in your final decision:

- **Issue the RFI** — Larger organizations should issue an RFI though established channels and contact a few leading SIEM/LM vendors directly. If you're a smaller organization, start by sending your RFI to a trusted VAR and email a few SIEM/LM vendors which seem appropriate for your organization.

- **Define the short list** — Before bringing anyone in, match any materials from the vendors or other sources against your RFI and draft RFP. Your goal is to build a short list of 3 products which can satisfy most of your needs. You should also use outside research sources (like Securosis) and product comparisons. Understand that you'll likely need to compromise at some point in this process, as it's unlikely any one vendor can meet every requirement.

- **Dog and Pony Show** — Instead of generic presentations and demonstrations, ask the vendors to walk you through specific use cases that match your expected needs. This is critical, because the vendors are very good at showing eye candy and presenting the depth of their capabilities, while redefining your requirements based on their strengths. Don't expect a full response to your draft RFP — these meetings are to help you better understand how each vendor can solve your specific use cases and to finalize your requirements.

- **Finalize and issue your RFP** — At this point you should completely understand your specific requirements, and issue a final formal RFP.

- **Assess RFP responses and start proof of concept (PoC)** — Review the RFP results and drop anyone who doesn't meet your hard requirements, such as platform support. Then bring in any remaining products for in-house testing. You'll want to replicate your projected volume and data sources if at all possible. Build a few basic policies that match your use cases, then violate them, so you can get a feel for policy creation and workflow. And make sure to do some forensics work and reporting so you understand the customization features. Understand that you need to devote resources to each PoC and stick to the use cases. The objective here is to put the product through its paces and make sure it meets your needs.

Selection and Deployment

- **Select, negotiate, and buy** — Finish testing, take the results to the full selection committee, and begin negotiating with your top two choices, assuming more than one meets your needs. Yes, this takes more time, but you want to be able to walk away from one of the vendors if they won't play ball with pricing, terms, and conditions.

- **Implementation planning** — Congratulations, you've selected a product, navigated the procurement process, and made a sales rep happy. But now the next stage of work begins — as the end of selection, you now need to plan the deployment. That means making sure of details like lining up resources, getting access/credentials to devices, locking in an install schedule, and even the logistics of getting devices to the right locations. No matter how well you execute on the selection, unless you implement flawlessly and focus on quick wins and getting immediate value from the SIEM/LM platform, your project will be a failure.

I can hear your groans from small to medium sized business who look at this process and think this is a ridiculous amount of detail. Once again we want to stress that we created a granular selection process, but you can pare this down to meet your organization's requirements. We wanted to make sure we captured all the gory details some organizations need to go through for a successful procurement. The process outlined is appropriate for a large enterprise but a little pruning can make it manageable for small groups. That's the great thing about process: you can change it any way you see fit at no expense.

Conclusion

For many organizations, implementing a Security Information and Event Management/Log Management (SIEM/LM) platform is a requirement, rather than a choice. Regardless of whether it's driven by compliance or operational issues, the impetus is on us as security professionals to get sufficient value from the technology. The sea of data continuously created by devices and applications is simply too vast to derive meaningful value from without automated assistance in gathering, normalizing, and providing basic correlation. We need to react to threats, but don't have weeks to sift through endless log files, and we need to *react faster* to have any hope of countering attacks. No, we don't pretend that SIEM can automate the skills of a seasoned security professional, but at least they can help filter the detritus out, and present the more relevant and interesting events to analysts and auditors for evaluation and possible followup. Quick and efficient handling of this data is critical for security and necessary to address compliance — and that is exactly what SIEM can provide if deployed wisely.

As we have discussed, the key is understanding why you are buying the technology and then building the selection/procurement process to meet those needs. That involves a clear understanding of the use cases, and requires spending sufficient time during the proof of concept stage to learn the tools and figure out which can satisfy all (or at least most) of the requirements. Sitting through a WebEx demo and placing an order is the wrong answer. Buying the 'class leading' product in the right quadrant and blindly assuming it will fit your needs is asking for disaster!

Selection only begins the process of getting value from your SIEM/LM platform. The deployment needs to be carefully planned to meet clear and articulated requirements, and the architecture designed for the size and organization of the purchasing entity. Success is attained by seizing quick wins — whether finding a compromised device or providing enhanced visibility into application or network behavior, it's critical to demonstrate value quickly. That will build momentum, which helps integrate the tool into daily work processes and builds confidence.

Most of all, make sure you are operationalizing the tool, which requires ongoing resources to keep the platform tuned, relevant, and complete. New devices and applications will be added, and those need to pump data into the SIEM/LM. New attacks will surface which you need to keep watching for. New data types will emerge which must be integrated into the tool. But none of this happens if you stop working once the initial deployment is complete. The environment is dynamic, and your tool needs to adapt as well. SIEM/LM is ***not*** a set-it-and-forget technology, and expecting the system to hum along without care and maintenance is a recipe for failure.

For those who make the choice to build a system and processes around the tool, huge efficiencies and improved security with quicker response to attacks are achievable. Our hope is that this guide provides the information necessary to evaluate products both individually and head-to-head, and help you avoid many of the problems that tend to pop up — usually just *after* purchasing a product, when coming to grips with how it *really* works. Good luck with your selection process — we are happy to help if you run into questions during your efforts; feel free to drop us a note at info@securosis.com, and we'll do our best to help out.

About the Analysts

Adrian Lane, Analyst/CTO

Adrian is a Security Strategist and brings over 22 years of industry experience to the Securosis team, much of it at the executive level. Adrian specializes in database security, data security, and software development. With experience at Ingres, Oracle, and Unisys, he has extensive experience in the vendor community, but brings a pragmatic perspective to selecting and deploying technologies — having worked on "the other side" as CIO in the finance vertical. Prior to joining Securosis, Adrian served as the CTO/VP at companies such as IPLocks, Touchpoint, CPMi and Transactor/Brodia. He has been invited to present at dozens of security conferences, contributed articles to many major publications, and is easily recognizable by his "network hair" and propensity to wear loud colors. Once you get past his windy rants on data security and incessant coffee consumption, he is quite entertaining.

Adrian is a Computer Science graduate of the University of California at Berkeley with post-graduate work in operating systems at Stanford University. He can be reached at alane (at) securosis (dot) com.

Mike Rothman, Analyst/President

Mike's bold perspectives and irreverent style are invaluable as companies determine effective strategies to grapple with the dynamic security threatscape. Mike specializes in the sexy aspects of security, such as protecting networks and endpoints, security management, and compliance. Mike is one of the most sought after speakers and commentators in the security business and brings a deep background in information security. After 20 years in and around security, he's one of the guys who "knows where the bodies are buried" in the space.

Starting his career as a programmer and a networking consultant, Mike joined META Group in 1993 and spearheaded META's initial foray into information security research. Mike left META in 1998 to found SHYM Technology, a pioneer in the PKI software market, and then held VP Marketing roles at CipherTrust and TruSecure — providing experience in marketing, business development, and channel operations for both product and services companies.

After getting fed up with vendor life, Mike started Security Incite in 2006 to provide a voice of reason in an over-hyped yet underwhelming security industry. After taking a short detour as Senior VP, Strategy and CMO at elQnetworks to chase shiny objects in security and compliance management, Mike joins Securosis with a rejuvenated cynicism about the state of security and what it takes to survive as a security professional.

Mike published *The Pragmatic CSO* <http://www.pragmaticcso.com/> in 2007 to introduce technically oriented security professionals to the nuances of what is required to be a senior security professional. He also possesses a very expensive engineering degree in Operations Research and Industrial Engineering from Cornell University. His folks are overjoyed that he uses literally zero percent of his education on a daily basis. He can be reached at mrothman (at) securosis (dot) com.

About Securosis

Securosis, L.L.C. <http://securosis.com> is an independent research and analysis firm dedicated to thought leadership, objectivity, and transparency. Our analysts have all held executive level positions and are dedicated to providing high-value, pragmatic advisory services.

Our services include:

- *Primary research publishing*: We currently release the vast majority of our research for free through our blog, and archive it in our Research Library. Most of these research documents can be sponsored for distribution on an annual basis. All published materials and presentations meet our strict objectivity requirements, and follow our Totally Transparent Research policy <http://securosis.com/about/totally-transparent-research>.
- *Research products and strategic advisory services for end users*: Securosis will be introducing a line of research products and inquiry-based subscription services designed to assist end user organizations in accelerating project and program success. Additional advisory projects are also available, including product selection assistance, technology and architecture strategy, education, security management evaluations, and risk assessments.
- *Retainer services for vendors*: Although we will accept briefings from anyone, some vendors opt for a tighter ongoing relationship. We offer a number of flexible retainer packages. Example services available as part of a retainer package include market and product analysis and strategy, technology guidance, product evaluations, and merger and acquisition assessments. Even with paid clients, we maintain our strict objectivity and confidentiality requirements.
- *External speaking and editorial*: Securosis analysts frequently speak at industry events, give online presentations, and write and/or speak for a variety of publications and media.
- *Other expert services*: Securosis analysts are available for other services as well, including Strategic Advisory Days, Strategy Consulting engagements, and Investor Services. These tend to be customized to meet a client's particular requirements.

Our clients range from stealth startups to some of the best known technology vendors and end users. Clients include large financial institutions, institutional investors, mid-sized enterprises, and major security vendors.

Additionally, Securosis partners with security testing labs to provide unique product evaluations that combine in-depth technical analysis with high-level product, architecture, and market analysis.