Security Management 2.0: Time to Replace Your SIEM?

Version 1.5
Released: October 24, 2011

Securosis, L.L.C. 515 E. Carefree Blvd. Suite #766 Phoenix, AZ 85085 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 but has been enhanced, reviewed, and professionally edited.

Special thanks to Chris Pepper for editing and content support.

Licensed by McAfee

McAfee, a wholly owned subsidiary of Intel Corporation (NASDAQ:INTC), is the world's largest dedicated security technology company. McAfee delivers proactive and proven solutions and services that help secure systems, networks, and mobile devices around the world, allowing users to safely connect to the Internet, browse, and shop the web more securely. Backed by its unrivaled global threat intelligence, McAfee creates innovative products that empower home users, businesses, the public sector, and service providers by enabling them to prove compliance with regulations, protect data, prevent disruptions, identify vulnerabilities, and continuously monitor and improve their security. McAfee is relentlessly focused on constantly finding new ways to keep our customers safe. http://www.mcafee.com

Copyright

Table of Contents

Introduction

Is it time? Are you waving the white flag? Has your SIEM failed to meet expectations despite your significant investment? If you are questioning whether your existing product or service can get the job done, you are not alone. You likely have some battle scars from the difficulty of managing, scaling, and actually doing something useful with SIEM. Given the rapid evolution of SIEM/Log Management offerings – and the evolution of requirements with new application models and this cloud thing – you *should* be wondering whether a better, easier, and less expensive solution meets your needs.

As market watchers, we don't have to be brain surgeons to notice the smaller SIEM and Log Management vendors innovating their various ways to relevance – with new deployment models, data storage practices, analysis techniques, and security features. Some vendors are actively evolving their platforms – adding new capabilities on top of what they have, evolving from SIEM features into broader *security management suites*. We note that others basically "milk the cash cow" by collecting maintenance revenue, while performing the technical equivalent of "putting lipstick on a pig". (Yes, we just used *two* farm animal analogies in one sentence.) You may recognize this phenomenon as the unified dashboard approach to hiding obsolescence. A plan that involves "buying another start-up with a shiny object!" ... to distract customers from a stagnant core product strategy.

> Don't assume that SIEM replacement is always the answer – that's simply not the case.

Let's face it – public company shareholders love milking cash cows while minimizing research and development costs. Security companies and their investors (especially those acquired by mega IT) have no problem with this model. Meanwhile customers tend to end up holding the bag, with few options for getting the innovation they need to keep pace with attackers. From our alarmingly consistent conversations with SIEM customers, we know it's time to focus on this dissatisfaction and open the SIEM replacement process up to public scrutiny. Don't be scared – in some ways SIEM replacement can be easier than the initial installation (yes, you can breathe a sigh of relief), but only if you leverage your hard-won knowledge and learn from your mistakes.

Security Management 2.0: Time to Replace Your SIEM? takes a brutally candid look at triggers for considering a new security management platform, walking through each aspect of the decision, and presenting a process to migrate – **if** the benefits outweigh the risks.

In this paper we will cover:

- **Platform evolution:** We will discuss what we see in terms of new features, platform advancement, and deployment models that improve scalability and performance. We'll also cover the rise of managed services to outsource the SIEM/Log Management functionality, and deploying hybrid configurations.

- **Requirements:** We'll examine the evolution of customer requirements in the areas of security, compliance, and operations management. We will also cover some common customer complaints, but to avoid devolving into a customer gripe session, we'll also go back and look at why some of you bought SIEM to begin with.
- **Platform evaluation:** We'll help walk through an in-depth examination of your current environment and its effectiveness. This will be a candid examination of what you have today – considering both *what works* and an objective assessment of what you're unhappy about.
- **Selection process:** This is an in-depth look at how to tell the difference between various vendors' capabilities, and at which areas are key for your selection. Every vendor will tell you they are "class leading" and "innovative", but most are not. We'll help you cut through the BS and determine what's what. We will also define a set of questions and evaluation criteria to help prioritize what's important and how to weigh your decision.
- **Decision process:** We will help re-evaluate your decisions by re-examining original requirements and helping remove bias from your consideration of shiny new features.
- **Negotiation:** You will be dealing with an incumbent vendor, and possibly negotiating with a new provider. We'll help you factor in the reality of your incumbent vendor's efforts try to save the business, and ways to leverage that as you move to solidify a new platform.
- **Migration:** If you are moving to something else, how do you get there? We'll provide a set of steps for migration, examining how to manage multiple products during the migration.

Don't assume that SIEM replacement is always the answer – that's simply not the case. In fact, after this analysis you may feel much better about your original SIEM purchase, with a much better idea (even a plan!) to increase usage and success. But you owe it to yourself and your organization to ask the right questions, and to do the work to get those answers. It's time to slay the sacred cow of your substantial SIEM investment, and figure out objectively what offers you the best fit moving forward.

SIEM Platform Evolution

Of the customers we talk with, there is general dissatisfaction with SIEM implementations – which in many cases have not delivered the expected value. The issues typically result from failure to scale, poor ease of use, challenges using the collected data in actionable timeframes, excessive effort for care & feeding and maintenance, or just customer execution failure. Granted, some of the discontent is clearly navel-gazing – parsing and analyzing log files as part of your daily job is boring, mundane, and error-prone work you'd rather not do. But dissatisfaction with SIEM is largely legitimate and has gotten worse, as system load has grown and systems have been subjected to additional security requirements, driven by new and creative attack vectors. This puts the spotlight on the fragility and poor architectural choices of some SIEM and Log Management platforms, especially early movers. Given that companies *need* to collect more data rather than less, review and management of the data just gets harder. Exponentially harder.

But let's not focus on user complaints – that doesn't help solve problems. Instead it makes mores sense to discuss the changes in SIEM platforms driving users to revisit their platform decisions. There are over 20 SIEM and Log Management vendors in the market, most of which have been at it for 5-10 years. Each vendor has evolved its products (and services) to meet customer requirements, and to provide some degree of differentiation against the competition. We have seen new system architectures to maximize performance, increase scalability, leverage hybrid deployments, and broaden data collection capabilities via standards (CEF) and universal collection format support. Usability enhancements include capabilities for data manipulation; addition of contextual data via log/event enrichment; and more powerful tools for management, reporting, and visualization. Data analysis enhancements include expansion of supported data types to include dozens of variants for monitoring, correlating/alerting, and reporting on change controls; configuration, application, and threat data; content analysis (poor man's DLP); and user activity monitoring.

With literally hundreds of new features to comb through, it's important to recognize that not all innovation is valuable to **you**, and you should keep irrelevancies out of your analysis of benefits of moving to a new platform. Just because the shiny new object has lots of bells and whistles doesn't mean they are relevant to your decision. Our research shows the most useful enhancements have been in scalability, along with reduced storage and management costs. Specific examples include mesh deployment models – where each device provides full logging and SIEM functionality – moving real-time processing closer to the event sources. The right architecture can deliver the trifecta of fast analysis, comprehensive collection/normalization/correlation of events, and single-point administration – but this requires a significant overhaul of early SIEM architectures. Every vendor meets the basic collection and management requirements, but only a few platforms do well at modern scale and scope.

With literally hundreds of new features to comb through, it's important to recognize that not all innovation is valuable to **you.**

These architectural changes to enhance scalability and extend data types are seriously disruptive for vendors – they typically require a "forklift upgrade": an extensive rebuild of the underlying data model and architecture. But the cost in

time, manpower, and disrupted reliability was too high for some early market leaders – so some instead *innovated* (yes folks, that's sarcasm) with sexy new bells and whistles, which were easier and faster to develop and show off, but left them behind the rest of the market in terms of real functionality. This is why we all too often see a web console, some additional data sources (such as identity and database activity data) and a plethora of quasi-useful feature enhancements tacked onto a centralized SIEM engine with limited scalability: that option cost less with less *vendor* risk. It is easy to be distracted from the most important SIEM advancements – those that **deliver on the core values of analysis and management at scale**.

Speaking of scalability issues, coinciding with the increased acceptance (and adoption) of managed security services, we are seeing many organizations look at outsourcing their SIEM. Given the increased scaling requirements of today's security management platforms, making compute and storage more of a service provider's problem is very attractive to some organizations. Combined with the commoditization of simple network security event analysis (for monitoring firewalls, IDS/IPS and servers), this has made outsourcing SIEM a reasonable alternative. Moving to a managed SIEM service also allows customers to *save face* by addressing the shortcomings of their current product without needing to acknowledge a failed investment. In this model, the customer defines the reports and security controls and the service provider deploys and manages SIEM functions.

Of course, there are limitations to managed SIEM offerings, so it all gets back to what problem you are trying to solve with your SIEM and/or Log Management deployment. To make things even more complicated, we also see hybrid architectures in early use, where a service provider does the fairly straightforward network (and server) event log analysis/correlation/reporting, while an in-house security management platform handles higher level analysis (identity, database, application logs, etc.) and deeper forensic analysis. We'll delve into the architectures in depth later in this paper.

Revisiting Requirements

Given the evolution of both the technology and the attacks, it's time to revisit your specific requirements and use cases – both current and evolving. You need to be brutally honest about what your existing product or service does and does not do, as well as your team's ability to support and maintain it. This is essential – you need a fresh look at the environment to understand what you **need** today and tomorrow, and what kind of resources and expertise you can bring to bear, unconstrained by what you do **today**. Many of you have laundry lists of things you would like to be able to do but can't with current systems. Wish lists are a good place to start, but you also need to consider your industry's trends, and look at what's coming down the road in terms of security and business challenges that will emerge over the next couple years. Our Security Management 2.0 process helps capture current and foreseeable needs.

Start with a blank slate

In order to figure out the best path forward for security management, we recommend you start with the proverbial blank slate. That means revisiting why you need a security management platform with fresh eyes. It means taking a critical look at use cases and figuring out their relative importance. As we described in Understanding and Selecting a SIEM/Log Management Platform, the main use cases for security management really break down into 3 buckets: Improving security, increasing efficiency, and automating compliance.

You need to be brutally honest about what your existing product or service does and does not do, as well as your team's ability to support and maintain it.

When you think about it, security success in today's environment comes down to a handful of key imperatives. First you need to **improve the security of your environment**. It's safe to say we have all lost ground to the bad guys, and we need to make some progress toward figuring out what's being attacked more quickly and protecting it. Unfortunately nobody's selling (working) crystal balls that tell us how and when attackers target us, so the blank slate strategy entails monitoring more and figuring out how your detection and response systems can react more quickly.

Next we need to do more with less. It does look like the global economy is improving but we can't expect to get back to the halcyon days of spend first, ask questions later – ever. There are specific automation and divide & conquer strategies that help reduce the burden. With more systems under management, we have more to worry about and less time to spend poring over reports, looking for the proverbial sharp pointy thing 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 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.

Increasingly technologies that monitor up the stack are helping in all three areas by collecting additional data types like identity, database activity monitoring, application support, and configuration management – as are different ways of addressing the problems. As attacks target these higher-level functions and require visibility beyond just the core infrastructure, the security management platform needs to detect attacks in the context of the business threat. Don't forget the need for advanced forensics, given the folly of thinking you can block every attack. So a security management platform to help React Faster and Better within an incident response context may also be a key requirement moving forward.

You might also be looking for a more integrated user experience across a number of security functions. For example, you may have separate vendors for change detection, vulnerability management, firewall and IDS monitoring, and database activity monitoring. You may be wearing out your swivel chair switching between all the consoles, and simplification via vendor consolidation might be a key driver.

Your general requirements may not have changed dramatically, although you may prioritize the use cases a little differently now. For example, perhaps you first implemented Log Management to crank out compliance reports. We see that as the primary driver in many cases. But you just finished cleaning up a messy security incident your existing SIEM missed. If so, you probably now put a pretty high value on making sure correlation works better.

Once you are clear within your team about the requirements for a security management team, start to discuss the topic a bit with external influencers. You should consult the ops teams, business users, and perhaps the general counsel about their requirements. Doing this confirms the priorities you already know and sets the stage to support a decision to move to a new platform — if you choose.

Critical evaluation (of yourself)

Now it's time to check your ego at the door. Unless you weren't part of the original selection team – then you can blame the old regime. Okay, we're kidding. Either way the key to this step is a brutally honest assessment of how your existing platform meets the needs that drove the initial implementation. This post-mortem type analysis evaluates the platform in terms of each of the main use cases (security, efficiency, and compliance automation), as well as some other aspects of real world use.

Even better, you'll need to determine why the product or service isn't measuring up. Common reasons we see include:

- **Ease of use:** Are there issues getting the product or service up and running? Did it require tons of professional services? Were you able to set up sufficiently granular rule sets and reports? This tends to be an issue with the product itself.
- **Implementation:** Were the rules configured correctly up front? Was the rule base maintained adequately as things changed, or was rule management so painful it tended to lag? Was all the proper data collected by the system to provide a broad view of your infrastructure? These issues tend to be *your* problems, and you need to own them. While deceiving yourself about how your organization implemented the technology might save a little face, it would only position you for another project failure.

- **Scalability:** Did your chosen platform just run out of gas when event volume ramped up? Did it take hours (or days) to run a report, forensics query, or fire a correlated alert? Were there architectural or even cost issues that prevented you from deploying a broader infrastructure to meet your needs? Did you have to surround the existing correlation engine with a set of logging devices to control event flow because the back end couldn't handle the volume? This might be a technology issue, or it could be a deployment architecture problem. Either way, the existing platform hasn't scaled to what you need, and that's a big issue.

- **Care and feeding:** Do you have adequate resources and expertise to optimize the system? Does keeping the back-end database operational require multiple FTEs? Has your staff been gutted to the point you don't have resources to monitor the system yourself? It's very important to realistically assess your team's ability to support the security management platform moving forward. The best technology in the world doesn't help much if you can't keep it up and running with a current rule set.

- **Forensics:** Does "drill down" mean manually looking through raw event logs? Worse, does it always involve going back to the archives to find the events you need? Were events normalized down to a useless subset of original data? Despite advancements in detection and alerting, forensic analysis is a common requirement for ascertaining the real severity of detected issues, and easier access to important data saves time and frustration.

- **Dying on the vine:** Has the technology been kept up to date? When was the vendor's last major update from the and did it address some of your issues? Has the vendor told you about the next release's road map? Have they made good on past promises of new capabilities? After big acquisitions, some products aren't maintained adequately (we know, that's a shock). Now you have to assess whether things will get better.

- **Vendor viability:** Did you buy a product from an early leader who has since hit hard times? Did their product roadmap involve driving off the road? Vendor fortunes can change dramatically after you buy their products, and you may need to reassess the vendor's ongoing viability. It's a bad day when you have to make a call to get source code delivered from escrow after creditors lock the vendor's doors.

Now you see why you need to check your ego at the door and make a brutally honest assessment of your team's ability to implement and support a security management platform. It would be great if this technology were plug and play, but it isn't. Regardless of whether you move to a new platform or not, you'll need to support it. It's very easy to just blame the vendor if the product hasn't met expectations, especially if the product has been left to die on the vine. But if there were implementation or maintenance issues on your side, those will still be there even with a modern, up-to-date platform. You can't blame the vendor for operational failure on your end.

If there were implementation or maintenance issues on your side, those will still be there even with a modern, up-to-date platform. *You can't blame the vendor for operational failure on your end.*

Now that you understand *what you need at this point* in time, and why your existing platform isn't meeting your needs, it's time to evaluate other options. Next we'll deal with new features available on these platforms and why some of the new capabilities are worth investigating.

Platform Evaluation

To understand the importance of picking a *platform*, as opposed to a product, for Security Management 2.0, let's draw a quick contrast between what we see when talking to customers of Log Management vs. SIEM. Most Log Management customers we speak with are relatively happy with their products. They chose a log-centric offering based on limited use cases – typically compliance-driven and requiring only basic log collection, simple analysis, and turnkey reporting. These products keep day-to-day management overhead low, and if they support the occasional forensic audit customers are generally happy. Log Management is an important – albeit basic – compliance tool. Think of it as like buying a can opener – it needs to perform a basic function and should always perform as expected. Customers don't want their can openers to sharpen knives, tell time, or let the cat out – they just want to open cans. It's not that hard. Log Management benefits from its functional simplicity – and even more from relatively modest expectations.

SIEMplicity (not so much)

Contrast that against conversations with SIEM customers. They have been at it for 5 years (maybe more), and as a result their installations have massive scope – in terms of both infrastructure and investment. They grumble about the tremendous growth in event collection driven by all these new devices. They need to collect nearly every event type, and often believe they need real-time response. The product had better be fast and provide detailed forensic audits. They depend on the compliance reports for their non-technical audience, along with detailed operational reports for IT. SIEM customers have a daily yin vs. yang battle between automation and generic results; between efficiency and speed; between easy and useful. It's like a can opener as one part of an entire machine shop, so everything is a lot more complicated. You can open a can, but first you might have to fabricate it from sheet metal.

It's important to understand that there are many moving parts in security management, and **setting appropriate expectations** is probably more important than any specific technical feature or function.

We use this overblown analogy because it's important to understand that there are many moving parts in security management, and setting appropriate expectations is probably more important than any specific technical feature or function. So your evaluation of a new platform needs to stay *laser focused* on the core requirements to be successful. In fact, the key to the entire decision-making process is understanding your requirements. We keep harping on this because it's the single biggest determinant of the success of your project.

When it comes to evaluating your current platform, you need to think about the issue from two perspectives. First, formally evaluate how well your platform addresses your current and foreseeable requirements in order to quantify critical features you depend on and identify significant deficiencies. Second, we will look at some of the evolving use cases and the impact of newer platforms on operations and deployment – both good and bad.

Just because another vendor offers more features and performance does not mean it's worth replacing your SIEM. *The grass is not always greener on the other side.* The former is critical for the decision process later in this series; the latter is critical for understanding the ramifications of a replacement decision.

Sizing up the incumbent

The first step in the evaluation process uses the catalog of requirements you built already to critically assess how the current SIEM platform achieves your needs. This means spelling out each business function, how critical it is, and whether the current platform gets it done. You'll need to discuss these questions with stakeholders from operations, security, compliance, and any other organizations that participate in the management of SIEM or take advantage of it. You cannot make this decision in a vacuum, and lining up support early in the process will pay dividends later on. Trust us on this.

Act like a detective, collecting these tidbits of information, no matter how small, to build the story of the existing SIEM platform in your environment.

Operations will be the best judge of whether the platform is easy to maintain and the complexity of implementing new policies. Security will have the best understanding of the product/service's forensic auditing capabilities, and compliance teams provide perspective on suitability of reports for audit preparation. Each audience provides a unique perspective on the criticality of each function and the effectiveness of the current platform.

In some cases you will find that the incumbent platform simply does not fill a requirement – that makes the analysis pretty easy. In other cases the system works perfectly, but is a nightmare in terms of maintenance and care & feeding for any system or rule changes. Performance may be less than ideal, but it's not clear what that really means, because the system could always be faster when investigating a possible breach. It may turn out the SIEM functions as desired but lacks capacity to keep up with all the events you need to collect, or takes too long to generate actionable reports. Act like a detective, collecting these tidbits of information, no matter how small, to build the story of the existing SIEM platform in your environment. This information will come into play later when you weigh options, and we recommend using a format that makes it easy to compare and contrast issues.

We offer the following table as an example of one method of tracking requirements, based on minimum attributes you should consider.

Requirement	Description	Criticality	Effectiveness	Notes
Compliance	PCI status reports	**Critical**	**Med**	*Lacks assessment & file access monitoring*
	Collect database administrator activity	**High**	**Low**	
	Need configuration baseline reports	**Low**	**N/A**	
Security	Identify NERC violations	**Critical**	**Med**	*Unable to construct policies*
	Provide real time views & analysis	**High**	**Low**	
Operations	Integration with federated identity systems	**High**	**Med**	*Active Directory is insufficient*
	Central policy management	**Med**	**N/A**	Central views, not management.
Misc	Virtual Appliance support	**Med**	**N/A**	

Security, compliance, management, integration, reporting, analysis, performance, scalability, correlation, and forensic analysis are all areas you need to evaluate in terms of your revised requirements. Prioritization of existing and desired features helps streamline the analysis. We reiterate the importance of staying focused on critical items to avoid "shiny object syndrome" driving you to select the pretty new thing, perhaps ignoring a cheap dull old saw that gets the work done.

Now we move the evaluation to other SIEM solutions. At this point in the process you have documented your requirements and *rationally* evaluated your current SIEM platform to determine what's working and what's not. This step is critical because a thorough understanding of your existing platform's strengths and weaknesses is the yardstick against which all other options will be measured. As you evaluate new platforms, you can **objectively** figure out whether it's time to move on and select another platform. Again, at this point no decision has been made. You are doing your homework – no more, no less.

You face two major difficulties during this phase of the process. First, you need to get close to some of the other SIEM solutions in order to dig in and determine what the other SIEM providers legitimately deliver, and what is marketing fluff. Second, you're not exactly comparing apples to apples. Some new platforms offer advantages because they use different data models and deployment options, which demands careful analysis of how a new tool can and should fit into your IT environment and corporate culture. Accepting some capabilities require you to push into new areas likely outside your comfort zone. We'll start by addressing common user complaints – and associated solutions – which highlight differences in function, architecture, and deployment.

The most common complaints we hear include: the SIEM does not scale well enough, users need more and better data, the product needs to be easier to use while providing more value, and users need to react faster to investigate the types of attacks happening today.

- **Scale:** With the ever growing number of events to monitor, it's simply not enough to buy bigger and/or more boxes to handle the exponential growth in event processing. Some SIEM vendors tried segregating reporting and alerting from collection and storage to offload processing requirements, which enables tuning each server to its particular role. This was followed by alternative deployment models, where log management collected the data (to meet scalability needs) and delivered a heavily filtered event stream to the SIEM to reduce its analysis load. But this is a band-aid, not a solution. New platforms address many of the architectural scaling issues, with purpose-built data stores providing fully distributed processing. These platforms can flexibly divide event processing/correlation, reporting, and forensic analysis. For more information on SIEM scaling architectures consult our Understanding and Selecting a SIEM/Log Management research.

- **Data:** Most platforms continue to collect data from an increasing number of devices, but many fail in two areas. First, they have failed to climb out of the network and server realm to start monitoring application assets in more depth. Second, many platforms suffer from over-normalization – literally normalizing the value right out of collected data. For many platforms, normalization is a necessary evil to address scalability limitations. This, coupled with poorly executed correlation and enrichment, produces data of limited value for analysis and reporting – which defeats the purpose. For example, if you need detailed information for business analytics, you'll need new agents on business systems – collecting application, file system, and database information *that is not included* in `syslog`. The format of this data is non-standard, and the important aspects of an application event or SQL query must be interpreted within the context of the application, requiring a deep understanding of what the application does and why. At times, you might feed these events through a typical data normalization routine and see nothing out of the ordinary. But if you examine the original transaction and dig into the actual query, you might find SQL injection (which is bad). Better data means both broader data collection options and more effective processing of the collected data within the context of the business process.

The tool was *supposed* to help – not create even more work.

- **Easier:** This fairly generic term encompasses several aspects: automation of common tasks, (real) centralized management, better visualization, and analytics. Rules that ship out of the box are traditionally immature (and mostly useless) as if written by tech companies with little understanding of your particular requirements — which they were. Automated reporting and alerting features got a black eye because they returned minimally useful information, requiring extensive human intervention to comb through thousands of false positives. The tool was *supposed* to help – not create even more work. Between better data collection, more advanced analytics engines, and easier policy customization, the automation capabilities of SIEM platforms have evolved quickly. Centralized management is not just a reporting dashboard across several unconnected products. We call that *integration on the screen*. To us, centralized management means analysis and reporting of events from across the enterprise, the ability to distribute rules from a central policy manager, **and** the capability to tune the rules on an enterprise basis. Most products *cannot* do this, but in distributed environments where you want to push processing closer to the point of attack you need. Useful visualization – not just shiny pie charts, but real graphical representations of trends, meaningful to the business – can help make decisions easier.

- **Speed:** Collection, moving the data to a central location, aggregation, normalization, correlation, and then processing is a somewhat antiquated SIEM model. Welcome to 2002. Newer SIEMs inspect events and perform some pre-processing *prior* to storage to ensure near-real-time analysis, as well as post-correlation analysis. These actions are computationally expensive, so recognize these advancements are predicated on an advanced product architecture and an appropriate deployment model. As mentioned in the data section, this requires SIEM deployment (analysis, correlation, etc.) to be pushed closer to the collector nodes – and in some cases even into the data collection agent.

Collection, moving the data to a central location, aggregation, normalization, correlation, and then processing is a somewhat antiquated SIEM model. Welcome to 2002.

Vendor Evaluation

Much of defining your evaluation criteria involves wading objectively through vendor hyperbole. As technology markets mature (and SIEM is pretty mature), the capabilities of each platform converge to a lowest common denominator. Similar messaging makes it increasingly hard to differentiate one platform from another. Given your unhappiness with your current platform (or you wouldn't be reading this, right?), it's important to distill what a platform does and what it doesn't, as early in the process as you can. And make no mistake, there are significant differences!

We divide the vendor evaluation process into two phases. First we'll help you define a short list of potential replacements. Maybe you use a formal RFP/RFI to cull the 25 companies in the space to 3-5, or maybe you don't. You'll see soon enough why you can't run 10 vendors through even the first stage of this process. At the conclusion of the short list exercise, you'll need to test one or two new platforms during a proof of concept, as we'll detail next. We don't recommend you skip directly to the test, by the way. Each platform has strengths and weaknesses and just because a vendor happens to be in the upper right quadrant of a magical chart doesn't mean it's the right choice for you.

Do your homework. All of it. Even if you don't feel like it.

Each platform has strengths and weaknesses and just because a vendor happens to be in the upper right quadrant of a magical chart doesn't mean it's the right choice for you.

Defining the short list

A few aspects of the selection criteria should be evaluated with a broader group of challengers. Think 3-5 at this point. You need to prioritize each of these areas based on your requirements. That's why you spent so much time earlier defining and gaining consensus on what's important for replacing your platform.

Your main tool at this stage of the process is what we kindly call the *dog and pony show*. That's when the vendor brings in their sales folks and sales engineers (SEs) to tell you how their product is awesome and will solve every problem you have. Of course, they won't be ready (unless they read this paper as well) for the *intensity* of your KGB-style interrogation techniques. Basically, you know what's important to you and you need confidence that any vendor passing through this gauntlet (and moving on to the PoC) will be able to meet your requirements.

Let's talk a bit about tactics to get the answers you need, based on the deficiencies in your existing product (from the platform evaluation). You need to get detailed answers at these meetings to be able to objectively evaluate any new platform. This meeting is not a 30 slide PowerPoint and a generic demo. Make sure the challenger understands those expectations **ahead** of the meeting, so they have the right folks in the room. If they bring the wrong people, cross them off the short list. It's as simple as that – it's not like you have a lot of time to waste, right?

We recommend defining a set of use cases/scenarios for the vendor to walk you through. Then the skilled folks with expertise using the tool can show you how they would solve the problem you have mapped out. This forces them to think about your problems rather than their scripted demo and enables you to really decipher the capabilities of the tool, instead of the expertise of the folks staging the demo.

- **Security:** The first scenario should focus on security. That's what this ultimately boils down to, right? You want to understand how they would detect an attack based on the information sources they gather and how they configure their rule sets and alerts. Make it detailed but not totally ridiculous. So basically simplify your existing environment a bit and run them through an attack scenario you've seen recently. This will be a good exercise for seeing how the data they collect solves a major use case, detecting an emerging attack quickly. Have the SE walk you through setting up and customizing a rule because you'll need to do both – often. Use your own scenario to reduce the likelihood of the SE having a pre-built rule. You want to really understand how the rules work, because you will spend a *lot* of time configuring your rules.

- **Compliance:** Next you need to understand what level of automation exists for compliance purposes. Ask the SE to show you the process of preparing for an audit. And no, showing you a list of 2,000 reports, most called PCI X.X, is not sufficient. Ask them to produce samples for a handful of critical reports you rely upon to see how closely they hit the mark – you can see the difference between reports developed by an engineer and those created by an auditor. You need to understand where the data comes from, and hopefully they have a demo data set to show you populated reports. The last thing you want is to learn that their reports don't pull from the right data sources two days before an audit.

- **Integration:** In this part of the discussion delve into how the product integrates with your existing IT stack. How does the platform pull data from your identity management system? CMDB? What about data collection? Are the connectors pre-built and maintained by the vendor? What about custom connectors? Is there a SDK available, or does it require a bunch of professional services? Or both? Buyer beware — don't leave any questions unanswered.

- **Forensics:** Vendors throw around the term *root cause analysis* frequently, while rarely substantiating how their tool works through an incident. Have the SE literally walk you through an investigation based on their sample data. Yes, you'll test this yourself later, but get a feel for the built-in tools and how they can be used by the SE, who should really know how to use the system.

- **Scalability:** If your biggest issue is a requirement for more power you'll want to know (at a very granular level) how each challenger solves the problem. Dive into their data model and their deployment architectures, and have them tell stories about their biggest implementations. If scalability is a problem for the incumbent you'll know how big the system needs to get and understand whether a proposed architecture passes the sniff test.

- **Additional data types:** What if you wanted to collect application data and build some rules? Have the SE walk you through the entire process: how they'd look at a data schema, set up the collector, and set up a few rules leveraging this new data. You aren't looking for empirical correctness, but you are evaluating whether unnatural acts are required to support additional data types.

Depending on your requirements and platform evaluation, there may be other areas you need to go through with each vendor.

Through each of these scenarios and areas to investigate, pay attention to user experience and ease of use. Is it 3 clicks to change a rule, or 10? Do you have to be a master of regular expressions to build and customize a rule or is there a visual rule builder? Can someone from the audit group understand and make changes, or do they need to be a network

admin to understand all the features? How do you set up a baseline in the system and then refine the alerting thresholds? Keep in mind the SE should be a ninja with their own product. At least the good ones are, so don't be afraid to ask them to slow down and show you *exactly* how the product works.

As you can see, this type of meeting could be considered cruel and unusual punishment in many countries. But it's critical that you get this level of detail **before** you commit to actually testing a product or service. Remember, this evaluation happened because the incumbent isn't getting it done. Shame on you if you don't ask every question you can to make sure the same mistakes aren't made again. Don't worry about making the SE uncomfortable – providing technical answers for you is their job.

Also don't expect to get through a meeting like this in 30 minutes. You will likely need a half-day (at minimum) to go through all these scenarios. That's why you probably want to only bring 3-5 vendors in for these meetings. You will be spending days with each product during proof of concept, so try to disqualify products that won't work *before* you spend that much time on them. This initial meeting can be a painful investment of time – especially if you realize early in the meeting that a vendor won't make the cut – but it's worth doing anyway. You'll thank us later.

After you finish your ritual humiliation of vendor sales teams, your gut will tell you which products or services are the right fit. But that's not good enough. You need to get hands on with the systems and run them through their paces for a couple days. That's the next step in the process and it will be different than your first Proof of Concept (PoC). You didn't know anything back then. Now you know what works – and more importantly what doesn't – and your evaluation will be better for it.

Driving the PoC

It's time to cull some vendors and create a short list based on your requirements, and to then move into the next step of the evaluation process – the PoC. Our PoC process is somewhat controversial – mostly because vendors hate it. Why? Because it's about **you** and your needs, not them and their product. But you are the buyer, right? Always remember that.

Most SIEM vendors want to push you through a 3-5 day evaluation of their technology on their terms, with their guy driving. You already have a product in place so you know the drill. You defined a few use cases important to you, and then the vendor (and their SE) stood the product up and ran through those use cases. They brought in a defined set of activities for each day, and you ended the test with a good idea of how their technology works, right?

Our PoC process is somewhat controversial – mostly because vendors hate it. Why? Because it's about **you** and your needs, not them and their product.

Wrong. The vendor PoC process is built to highlight their product strengths and hide their weaknesses. We know this from first hand experience – we have built them for vendors in past lives. You need to work through **your** situation, not theirs. Find the warts now – **not** when you are responding to an incident. It's wacky that some vendors get scared by a more open PoC process, but their goal is to win the deal, and they put a lot of sweat into scripting their process so it goes smoothly. We hate to say it, but smooth sailing is not the point! The vendor will always say "We can do that!" – it's your job to find out how well – or how awkwardly.

Before you start a PoC establish evaluation criteria based on your requirements and use cases. Your criteria don't need to be complicated. Your requirements should spell out the key capabilities you need, with a plan to further evaluate each

challenger based on intangibles such as set-up/configuration, change management, customization, user experience/ease of use, etc. Before you start, have your team assess your current platform against the same criteria as a baseline for comparison.

As you start the PoC, we recommend you invest in screen capture technology. It's hard to remember what these tools did and how they did it – especially after you've seen a few of them work through the same procedures. So capture as much video as you can of the user experience – it will come in very handy when you get to the decision point.

Without further ado, let's jump into the PoC.

Stand it up for real

One of the advantages of testing security management products is that you can actually monitor production systems without worrying about blowing them up, taking them down, or adversely impacting anything. So do just that. Plan to pull data from your firewalls, your IDS/IPS systems, and your key servers. Not all devices, of course, but enough to get a feel for how you need to set up the collectors. You will also want to configure a custom data source or two and integrate with your directory store to see how that works. Actually do a configuration and bootstrap the system in your environment.

Keep in mind that the PoC provides a great opportunity to get some professional services help – gratis. This is part of the sales process for the vendors, so if you want to model out a targeted attack and then enumerate the rules in the system, have the SE teach you how. Then model out another attack and build the rules yourself, without help. The key is that your team learns how to run the system and gets comfortable – if you do switch you will be living with your choice for a long time.

Focus on visualization, your view into the system. Configure some dashboards and see the results. Mess around with the reports a bit. Tighten the thresholds of the alerts. Does the notification system work? Will the alerts be survivable at production levels for years? Is the information useful? These are all things you need to do as part of kicking each challenger's tires.

If compliance is your key requirement use PCI as an example. Start pulling data from your protected network segment. Pump that data through the PCI reporting process. Is the data correct and useful for everybody with an interest? Are the reports comprehensive? Will you need to customize the reports for any reason? You need to answer this kind of questions during the PoC.

Run a Red Team

Run a simulated attack against yourself. We know actually attacking production systems would make you very unpopular with the ops folks, so set up a lab environment. But you want as realistic a situation as possible. Have attackers breach test systems with attack tools. Have your defenders try to figure out what is going on as it's happening. Does the system alert as it should? Will you need to heavily customize the rule set? Can you identify the nature of the attack quickly? Does their super-duper forensic drill-down give you the view you need? The clock is ticking, so how easy is it to use the system to search for clues?

Obviously this isn't a real incident situation, so you'll take some editorial liberties, and that's fine. You want a feel for how the system performs in near-real-time. If an attacker is in your systems, will you find them? In time to stop or catch them? Once you know they are there, can you tell what they are doing? A Red Team PoC will help you determine that.

Do a post-mortem

Once you are done with the Red Team exercise, you should have a bunch of data that will make for a nice forensic investigation of what the attack team did, and perhaps what the defense team didn't do as well as they could have. This is a learning experience for everyone. Will the tool hold up in the heat of battle? How does it compare to your existing product for comparable functions?

You cannot possibly prevent all attacks from succeeding, so you need practice on your incident investigation and response processes. This type of simulation forces you to exercise facets of the product you might otherwise miss.

As important is the experience of running a simulated attack on your team. You cannot possibly prevent all attacks from succeeding, so you need practice on your incident investigation and response processes. This type of simulation forces you to exercise facets of the product you might otherwise miss.

You can't fully test scalability during the PoC so focus on the stuff you can see, feel, and touch. That's the user experience, and there is no better way to distill out the effectiveness of each challenger than to stage an attack. Remember to have your team grade the challenger while their memory is fresh and their perceptions are raw. After spending 1-2 weeks with another product, they won't remember what they liked and what they didn't – which is where the screen grabs come in handy.

Lather, Rinse, Repeat

You will probably test more than one product or service, so you get to do this all again. Given the resource-intensive nature of this testing process, you probably cannot put more than 2 products through a comprehensive PoC, but do use the same scenarios for each product. That consistency helps make the challenge fair and your comparison more meaningful.

Now you have all the information you need to make a decision, it's time to figure out what to do and gather data to substantiate your choice. You can use the grades and videos you collected for each challenger – especially in making the case for a new platform, if that's what you decide on. See? There is some method to our madness.

Making Your Decision

It's time – you are ready. Now it's time to make the call. We know the importance of this decision – you're here because your first attempt at this project wasn't as successful as it needed to be. So let's break down the decision to ensure you can make a good recommendation and feel comfortable with it.

That's actually a good point to discuss. The output of our Security Management 2.0 process is not really a *decision* – it's more of a recommendation. The final decision will likely be made in the executive suite. That's why we have focused so much on gathering data (quantitative where possible) – you will need to defend your recommendation until the purchase order is signed. And probably afterwards.

We won't mince words. This decision generally isn't about the facts – especially since there is an incumbent in play, which may be part of a big company that has important relationships with heavies in your shop. So you need your ducks in a row and a compelling argument for any change. And even then, you may not be able to push through a full replacement. In that case, the answer may be to supplement. In this scenario, you still aggregate information with the existing platform, but then feed it to the new platform for analysis, reporting, forensics, etc. across the enterprise. Again, given the economic investment of running both, this may not be palatable for some organizations, but if your hands are tied relative to replacement, this kind of creative approach is worth considering.

But that's still only the external part of the decision process. In many cases, the (perceived) failure of your existing SIEM may be self-inflicted. So we also need to evaluate and explain the causes of the failed project, with assurance that you can avoid those issues *this time*. If not your successor will be in the same boat in another 2-3 years. So before you put your neck on the chopping block and advocate for change (if that's what you decide), do some deep internal analysis as well.

Introspection

First let's make sure you *really* re-examined the existing platform in terms of the original goals. Did your original goals adequately map your needs at the time, or was there stuff you did not expect? How have your goals changed over time? Be honest! Do not let your ego get in the way of doing what's right, and take a hard and fresh look at the decision to ensure you don't repeat previous mistakes. Did you kick off this process because you were pissed at the original vendor? Or because they got bought and seemed to forget about the platform? Do you know what it will take to get the incumbent where it needs to be – and whether that is even possible? Is it about throwing professional services at the issues? Is there a fundamental technology problem?

Remember, there are no right or wrong answers here, but the truth (and your commitment) will become clear when you need to sell this to management. Some of you may worry management will look at the need for replacement as 'your fault' for choosing the incumbent, so make sure you have answers to these questions and that you aren't falling into a self-delusion trap. You need your story straight and your motivations clear.

Did you assess the issues critically the first time around? If it was a skills issue, have you addressed it? Can your folks build and maintain the platform moving forward? Are you looking at a managed service to take that concern off the table? If it was a resource problem, do you now have enough staff for proper care and feeding? Yes, the new generation of platforms requires less expertise to keep operational, but don't be naive – no matter what any sales rep says, you cannot simply set and forget them. **Whatever** you pick will require expertise to deploy, manage, tune, and analyze reports. These platforms are not self-aware – not by a long shot.

The last thing you want to do is set yourself up for failure, so make sure you ask the right questions ahead of time and be honest about the answers.

> Some of you may worry management will look at the need for replacement as 'your fault' for choosing the incumbent, so make sure you have answers to these questions and that you aren't falling into a self-delusion trap.

Expectations

Next you have to reconcile expectations with reality. Revisiting requirements provides information on what you *need* the security management platform to do. You can prioritize specific use cases (compliance, security, forensics, operations), and have a pretty good feeling about whether the new platform or incumbent will meet your expectations. Remember, not everything is Priority #1, so pick your top three must-have items and prioritize the requirements.

If you love some new features of the challenger(s), will your organization leverage them? Firing off alerts faster won't help if your team takes a week to investigate each issue, or cannot keep up with the increased demand. The new platform's ability to look at application and database traffic doesn't matter if the developers won't help you understand normal behavior to build the rule set. Fancy network flow analysis can be a productivity sink if your DNS and directory infrastructure is a mess and you can't reliably map IP to user ID.

Does your existing product have too many features? Yes, it does happen that some organizations simply cannot take advantage of (or even handle) complex multi-variate correlation across the enterprise. Do you need to aggregate logs because organizational politics, or your team's resources or skill set, prevent you from getting the job done? This might be a good reason to outsource or use a managed service. There isn't a right or a wrong answer here. And not being honest about that answer will land you in the hot seat *again*.

If you kickstarted this effort because the existing SIEM missed something and it resulted in a breach, can you honestly say the new thing *would* (not 'might') detect that attack? We have certainly seen high profile breaches result in tossing the old and bringing in the new (someone has to pay, after all), but make sure you address the root cause of the **real** problem. And make sure you evaluate the right tool. If you were compromised by a persistent attacker, you may be looking at the wrong technology. Maybe you really need full packet capture. Maybe not, but *at least ask the question*.

> If you kickstarted this effort because the existing SIEM missed something and it resulted in a breach, can you honestly say the new thing *would* (not 'might') detect that attack?

Is is realistic to think you can deploy a common rule set across your enterprise? We talk about a central management and reporting capability, which doesn't matter if every operating division maintains its own IT shop with its own monitoring tools and it takes an act of Congress to get them in the same room. Don't expect people (or politics) to change just because the organization gains the ability to monitor across the enterprise. Having a technical capability doesn't mean you will have agreement on whether or how to use it. Don't forget about the political realities of your organization as well. Will people regard this whole project as snooping or unwelcome interference? That might impact project success, and so requires active management of expectations.

Economics

Even if your introspection and expectations are in line, there is still economic reality. IT groups need to *do more with less*, and that's not changing. So you need to weigh the cost of getting your existing product functional against replacing it. Even if you want to stack the deck against the incumbent, don't forget the cost (and time) to train your folks on the new tool – or the professional services to migrate your existing rules, data sources, and data to the new tool. If you managed to get some of that done during the proof of concept, that's great, but there's certain to be more as you move to full deployment.

Be sure you compare apples to apples. Sure the new platform may do a lot more, but at what cost? If essential reports do not get produced, or you find yourself running on what feels like a deprecated platform, you still need to account for the purchase, maintenance, deployment, customization, and training of the new thing. Make sure you consider **total** instead of merely **procurement** cost. Even OpEx is real money, at least the last time we checked.

Documentation

The end goal is a recommendation, so you need to document what you think and then present it to the folks with the money. Given all we know, here is how we would structure that artifact of your decision process:

- **Requirements:** Tell them what you need, and who told you that you need it. Compliance and security requirements come from different groups, so make sure these dependencies are referenced.
- **Current product evaluation:** What works and what doesn't work with your existing solution *within the context of your requirements*, both now and for the next set of use cases.
- **Challenger assessment:** Summarize the work you did to find the next platform. Which vendors did you disqualify and why – you never know if your CFO's brother-in-law works for a vendor. What did you learn in the proof of concept? Which competitor came out on top?
- **Cost estimate:** What would it cost to move to the new platform? What is capital expense and what is operational? What kind of investment in professional services would be required? How does that differ from making incremental improvements to the incumbent, and what functionality are you sacrificing in the move?

- **Migration plan:** If you ultimately decide to replace the SIEM, what does the migration look like? How long will it take? Will there be a disruption of service? Will there be any exposures and for how long? You need all these answers before you pitch to the powers that be. Not to the point of building a Gantt chart – that comes at the end – but enough to answer the tough questions.
- **Recommendation:** Your entire document should be building to this point, where you put the best path down on paper. If it's a surprise to your audience, you did something wrong. This is about telling them what they already know, and making sure they have an opportunity to ask any more questions.

But even once the recommendation is in, the decision is far from done. Now you have to negotiate with the new vendor (or the incumbent) and this is when your process is most vulnerable. An incumbent losing the deal will act desperately to save it. Challengers will do likewise if they think you're staying put or choosing a competitor.

Negotiation

You made your decision and sent it up the food chain, so now the fun begins. Well, fun for some folks, anyway. For this discussion we'll assume you have decided to move to a new platform. We understand some people decide not to move, but use the possibility switch for leverage in negotiations. But it bears repeating that it is not a bad thing to stay with your existing platform, so long as you have done the work to determine it can meet your requirements. We're writing this paper for the people who keep telling us about their unhappiness, and how their evolving requirements have not been met. So after asking all the right questions, if the best answer is to stay put, that's a less disruptive path anyway.

Replacement tactics

For now, though, let's just assume the current platform won't get you there. Now your job is to get the best price for the new offering. Here are a few tips to leverage for the best deal:

- **Time the buy:** Yes, this is Negotiation 101. Wait until the end of the quarter and squeeze your sales rep for the best deal to get the PO in by the last day of the month. Sometimes it works, sometimes it doesn't. But it's worth trying. Understand the rep may ask for your commitment that the deal will, in fact, get done that quarter. Make sure you can get it done if you pull this card.

- **Tell the incumbent they lost the deal:** Next get the incumbent involved. Once you put in a call letting them know you are going in a different direction they usually respond. Not always, but most times the incumbent will try to save the deal. And then you can go back to the challenger and tell them they need to do a little better, because you got this great offer from their entrenched competition. And just like when buying a car, to use this tactic you must be willing to walk away from the challenger and stay with the incumbent.

- **Look at non-cash add-ons:** Sometimes the challenger can't discount any more. But you can ask for additional professional services, modules, boxes, whatever. Remember, the incremental cost of software is zero, zilch, nada – so vendors can often bundle in a little more to get the deal when pushed to the wall.

- **Revisit service levels:** Another non-cash sweetener could be an enhanced level of service. Maybe it's a dedicated project manager to get your migration done. Maybe it's the Platinum level of support, even if you pay for Bronze. Given the amount of care and feeding required to keep any security management platform tuned and optimized, having a deeper service relationship could come in handy.

- **Dealing with your boss's boss:** One last thing – be prepared for your recommendation to be challenged, especially if the incumbent sells a lot of other gear to your company. This entire process has prepared you for that call, so just go through the logic of your decision once more, making clear that your recommendation is the best direction for the organization. But expect the incumbent to go over your head – especially if they sell a lot of storage or servers to your company.

Tactics for the incumbent

It would be pretty naive to not prepare in case the decision goes the other way – due to pricing, politics, or any other reason beyond your control. So if you have to make the status quo work and keep the incumbent, here are some ideas for making lemonade from the proverbial lemon:

It would be pretty naive to not prepare in case the decision goes the other way – due to pricing, politics, or any other reason beyond your control.

- **Tell the incumbent they're losing the deal:** We know it's not totally above board but all's fair in love, war, and sales. If the incumbent didn't already know they were at risk, it can't hurt to tell them. Some vendors (especially the big ones) don't care, which is probably why you looked at new stuff anyway. Others will get the wake-up call and try to make you happy. That's the time to revisit your platform evaluation and figure out what needs fixing.

- **Get services:** If your issue is not getting proper value from the system, push to have the incumbent provide some professional services to improve the implementation. Maybe send your folks to training. Have their team set up a new set of rules and do knowledge transfer. There are many options, but if you have to make do with what you have, at least force the vendor's hand to make the systems work better. We've seen examples of an organization literally starting over, which may make sense if your initial implementation is that screwed up.

- **Scale up (at lower prices):** If scalability is the issue, confront that directly with the incumbent and request additional hardware and/or licenses to address the issue. Of course this may not be enough but every little bit helps, and if moving to a new platform isn't an option, at least you can ease the problem a bit. Especially when the incumbent knows you were looking at new gear because of a scaling problem.

- **Add use cases:** Another way to get additional value is to request additional modules be thrown into a renewal or expansion deal. Maybe add the identity module or look at configuration auditing. Or work with the team to add database and/or application monitoring. Again, the more you use the tool, the more value you'll get, so figure out what the incumbent will do to make you happy.

Honestly, if you **must** stick with the existing system, you don't have much flexibility. The incumbent doesn't need to know that, though, so try to use the specter of migration as leverage. But at the end of the day, it is what it is. Throughout this process you have figured out what you need the tool to do, so now do your best to get there, within your constraints.

Once the deal is done, it's time to move to the new platform. We will wrap up this paper by discussing migration and helping plan to get onto the new kit. It will be hard – it always is – but you can leverage everything you learned through your first go-round with the incumbent, as well as this process, to build a very clear map of where you need to go and how to get there.

Migration

At this point we have outlined a disciplined and objective process to determine whether it's worth moving to a new security management platform. Assuming your organization decides to move, now it gets real. You need to implement and migrate your existing environment to the new thing, while maintaining service levels and without opening your organization to any additional risk. Walk in the park, right? Let's address these migration issues: the following is a personal trial in system migration, so hopefully you can learn from my pain.

Adrian started work at a previous employer two days after an IT consultancy performed a server migration. Coincidentally, at the same time he was helping a friend at a major bank review his data center migration plans. The bank had every phase of the change-over planned down to half-day increments, with backup plans in place, following months of migration rehearsals. Let's just say the IT consultancy had less elaborate plans. Bank employees knew their systems were critical and treated the migration as such – IT consultants, not so much. When Adrian walked into the offices at his new job every server was down, removed from the racks, sitting in a pile by the door. The consultancy was *assembling* the new hardware – and had been for more than a day. Their plan was to finish the hardware in a day or so; then they would install the operating systems. Once they had the identity management system working, they planned to install applications and import customer data. Out in the hallway, a few dozen *very* angry sales people paced the halls, idle, 3 weeks before the close of the quarter. It was a bad day for everyone.

Shockingly enough, the IT consultancy's contract was terminated that day. After plugging the old servers back in and dispersing the lynch mob outside the server room, Adrian planned out how to migrate to the new servers without any additional downtime. It was not just for the business's sake, but to ensure his personal safety as well. While he did not go to the same extremes as my friend's team at a certain giant, he acknowledged the servers were no less critical to his business, and a seamless migration of services was mandatory.

A flash cutover never really is. We recommend you start deploying the new SIEM long before you get rid of the old.

What can we learn from this somewhat transformative experience? A flash cutover never really is. We recommend you start deploying the new SIEM long before you get rid of the old. At best, you'll deprecate portions of the older system after newer replacement capabilities are online, but you will likely want the older system as a fallback until the new functions have been vetted and tuned. We have learned the importance of this staging process the hard way. Ignore it at your own peril, keeping in mind that your security management platform sustains several key business functions.

We have broken the migration process into two phases: planning and implementation. Your plan needs to be very clear and specific about when things get installed, how data gets migrated, when you cut over from the old systems to the new, and who performs the work.

Plan

The Planning step leverages much of the work done up to this point in evaluating replacement options – you just need to adapt it for the migration.

- **Review:** Go back through the documents you created earlier. First consider the platform evaluation documents, which will help you understand what the current system provides and key deficiencies to address. These documents become the priority list for the migration effort, and basis for the migration task list. Next, leverage what you learned during the PoC. When evaluating your new security management platform provider you conducted a mini-deployment exercise. Use what you learned from that exercise – particularly what worked and what didn't – as input for subsequent planning and address the issues it identified.

- **Focus on incremental success:** What do you install first? Do you work top down or bottom up? Do you keep both systems operational throughout the entire migration or do shut down portions of the old as each node migrates? We recommend using your deployment model as a guide. You can learn more about these models by checking out Understanding and Selecting a SIEM. When using a *mesh* deployment model, it's often easiest to make sure a single node/location fully functions before moving on to the next. With *ring* architectures it's generally best to get the central SIEM platform operational, and then gradually add nodes around it until you reach the scalability limit of the central node. Hierarchal models are best deployed *top-down*, with the central server first, followed by regional aggregation nodes in order of criticality, then down to the collector level. Break the project up to establish incremental successes and avoid dead ends.

- **Allocate resources:** Who does the work? When will they do it? How long will it take to deploy the platform, data collectors, and/or log management support system(s)? This is also the time to *engage professional services* **and** enlist the new vendor's assistance. The vendor presumably does these implementations all day long, so they should have expertise at estimating these timelines. You may also want to engage them to perform some (or all) of the work in tandem with your staff, at least for the first few locations, until you get the process down.

The vendor presumably does these implementations all day long, so they should have expertise at estimating these timelines.

- **Define the timeline:** Estimate the time it will take to deploy the servers, install the collectors, and implement your policies. Include time for testing and verification. There is likely to be some 'guesstimation' on your part, but you have some reasonable metrics to base your plan on, from the PoC and prior experience with SIEM. You did document the PoC, right? Plan the project commencement date and publish to the team. Solicit feedback and adjust before commencing, because you **need** shared accountability with the operations team(s) to make sure everyone has a vested interest in project success.

- **Preparation:** We recommend you do as much work as possible before you begin migration, including construction of the rules and policies you will rely on to generate alerts and reports. Specify in advance any policies, reports, user accounts, data filters, backup schedules, data encryption, and related services you can. You already have a rule base, so leverage it to get going. Of course you'll tune things as you go, but why reinvent the wheel or rush unnecessarily? Keep in mind that you will *always* find something you failed to plan for – often an unexpected problem – that sets your schedule behind. Preparation helps spot missing tasks and makes deployment go faster.

Implement

Remember that the migration need not (and in fact generally should not) be an all-at-once exercise – you have the luxury of doing one piece at a time in the order that best suits your requirements.

- **Deploy platform(s):** This varies based on the deployment model, as discussed above, but typically you install the main security management platforms first. Basic system configuration, identity management and access control integration, and basic network configuration. Once complete, connect to a couple data sources and other aggregation points to make sure the system is operating correctly.

- **Deploy supporting services:** Deploy the data collectors and make sure event collection is working correctly. If you use a flat deployment model configure the platform to collect these events for the first set of deployment tasks. If you use a Log Management/SIEM hybrid or regional data aggregators, install those additional aggregation points and get them feeding data into the primary SIEM system to confirm proper information flow – at a small scale before ramping up event traffic. If you are moving to a new platform to get real-time analysis, make sure event collection happens properly. The only concern should be getting data into the system in a timely fashion right now – tune it later.

- **Install policies and reports:** Next deploy the rules that comb through events and find anomalies. Hopefully you created as many as possible during the PoC and planning stages, and perhaps you can leverage your initial implementation. For real-time analysis you need to tune those rules to optimize performance. Remember each additional rule added incurs a significant processing cost. It's a math thing, as correlating multiple data sources against many rules causes the system to do exponentially more work, thus reducing effective performance and throughput. Look for ways to create rules with fewer comparisons, and balance fine-tuning rules to specific problems against more generic rules that catch many problems – sometimes you can throw hardware at the problem (with a bigger server) to handle more events, but it's always useful to strive for more efficient policies.

- **Test and verify:** Are your reports being generated properly? Are the correct alerts being generated in a timely fashion? Generate copies of the reports and send them to the team for review, and compare against the existing platform (which is still operational, right?). For alerts and forensic analysis, it makes sense to rerun your "Red Team" drill from the PoC to make sure you catch anomalies and confirm the accuracy of your results. Verify you get what you need – *now* is the time to find any problems with the system — while you still have a chance to find and fix problems and before you start depending on the new platform.

- **Stakeholder sign-off:** Get it in writing – trust us, this will save aggravation in the future when someone from Ops says: "Hey, where is XYZ that used to be there?" Have the compliance, security, and IT ops teams sign off on completion of the project – they own it now too (remember shared accountability?). Make sure the group is satisfied and/or all issues are documented – if not fully solved – by this point.

- **Decommission:** Now you can retire the older system. You may choose to run the incumbent SIEM for a few months after the new system is fully operational, just in case. But there are not many reasons to keep the older system around, and plenty of reasons it should be sent packing. Older agents and sensors should be removed, user accounts dedicated to the older platform locked down, and hardware and virtual server real estate reclaimed. Once again, someone will need to be assigned the work with an agreed-on time frame for completion. Trouble-ticketing systems are a handy way to schedule these tasks and provide automated completion reports.

Conclusion

So with all that we finish our analysis of Security Management 2.0. Did you decide it was time for a new SIEM? Let's revisit some of the critical aspects:

- **Requirements rule:** Take this opportunity to figure out what you *really* need – not what the vendor says you can do, or what your users read in a trade rag. Defining your requirements is the linchpin of this entire process, so make sure you do that well.
- **Do the work:** When a project is perceived as a failure, the inclination is to just change in hopes the new thing will be better. Ultimately that might be the right answer, but don't embark on such a major project based purely on a blind optimism. Evaluate your platform objectively and assess the challengers skeptically. Everything looks great in a PowerPoint deck, but leverage the PoC to see what will really work in your environment.
- **Be creative**: In the event you may not be able to totally replace your incumbent, you may need to think a bit out of the box to address the capability gaps of your existing SIEM. You may want to consider supplementing it with either a different platform or a managed service. It's wise to stay focused on the problems you need to solve (defined when revisiting your requirements) and using those requirements to make a compelling case as to why investment in supplemental technology (or services) is needed.
- **Be inclusive:** Any security management technology needs to be leveraged across the organization – if only for dashboards and reports. So make this process as inclusive as possible. That means getting buy-in from not just the senior team and the money folks, but also from operations and administrators. If you plan to capture data from application and database sources, include those teams in the process. You need their help, and want them to share responsibility for making the project succeed.
- **Get quick wins:** Focus on achieving consistent successes. That means starting slowly on areas you know will work well. Get one thing done correctly before moving on to the next one. Remember, most likely this whole process stems from an issue perceived with the incumbent, so make sure the new tool will work well, which requires finishing what you start.

If you have any questions on this topic, or want to discuss your situation specifically, feel free to send us a note at info@securosis.com.

About the Authors

Adrian Lane, Analyst and CTO

Adrian Lane is a Senior Security Strategist with 24 years of industry experience, bringing over a decade of C-level executive expertise to the Securosis team. Mr. Lane specializes in database architecture and data security. With extensive experience as a member of the vendor community (including positions at Ingres and Oracle), in addition to time as an IT customer in the CIO role, Adrian brings a business-oriented perspective to security implementations. Prior to joining Securosis, Adrian was CTO at database security firm IPLocks, Vice President of Engineering at Touchpoint, and CTO of the secure payment and digital rights management firm Transactor/Brodia. Adrian also blogs for Dark Reading and is a regular contributor to Information Security Magazine. Mr. Lane is a Computer Science graduate of the University of California at Berkeley with post-graduate work in operating systems at Stanford University.

Mike Rothman, Analyst and 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 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 executive roles at CipherTrust and TruSecure. 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 at eIQnetworks to chase shiny objects in security and compliance management, Mike joined 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. 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:

- The Securosis Nexus: The Securosis Nexus is an online environment to help you get your job done better and faster. It provides pragmatic research on security topics that tells you exactly what you need to know, backed with industry-leading expert advice to answer your questions. The Nexus was designed to be fast and easy to use, and to get you the information you need as quickly as possible. Access it at https://nexus.securosis.com.
- 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 conform to our Totally Transparent Research policy.
- 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 assessment.
- 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. Services available as part of a retainer package include market and product analysis and strategy, technology guidance, product evaluation, and merger and acquisition assessment. Even with paid clients, we maintain our strict objectivity and confidentiality requirements. More information on our retainer services (PDF) is available.
- 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. For more information about Securosis, visit our website: <http://securosis.com/>.