Software vs. Appliance: Database Activity Monitoring Deployment Tradeoffs

Version 1.0
Released: June 1, 2011

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 Application Security Inc.

About AppSec:

Founded in 2001, Application Security, Inc. (AppSec) has pioneered database security, risk, and compliance solutions for the enterprise. AppSec empowers organizations to assess, monitor, and protect their most critical database assets in real time, while simplifying audits, monitoring risk, and automating compliance requirements.

As the leading provider of cross platform solutions for the enterprise, AppSec's products – AppDetectivePro for auditors and IT advisors, and DbProtect for the enterprise – deliver the industry's most comprehensive database security solution. With over 2,000 customers in 42 countries, AppSec is headquartered in New York City and has offices throughout North America and the United Kingdom.

For more information, please visit www.appsecinc.com.

Copyright

Introduction

Appliance vs. Software: Which is better? With Database Activity Monitoring, the appliance vs. software debate is still raging. It's not front and center in product marketing materials, nor presented as a core value for solving security challenges, but positioned by vendors as a critical competitive advantage. Behind the scenes — especially during vendor bake-offs — deployment models are used to differentiate products. Vendors tend to focus their criticism not on the ways events are processed, UIs, or event storage, but on deployment models instead. Hardware is *better* than software. Software is *better* than hardware. This virtual appliance is *just as good* as software. And so on.

To customers new to Database Activity Monitoring, the advantages and disadvantages of each option are not readily apparent. Vendors make various evangelical pitches touting *the right way* to tackle monitoring, but customers cannot validate such claims until *after* they have purchased a solution and placed it in production. Many problems — and even many advantages — inherent to each model are not apparent until a year or more after deployment. At that point it's a bit late to tell the vendor you think they misstated the facts, as they have already cashed your check.

This paper examines into the practical differences between appliance, software, and virtual appliance based Database Activity Monitoring systems. Our goal is to assist customers in choosing a system, based on the short-term and mid-term tradeoffs of each deployment model, and help navigate through vendor rhetoric. I'll share some personal experiences — having worked in this market for the last decade — that illustrate differences between the theoretical and the practical. The research is presented in four parts, each focusing on a particular deployment topic.

Deployment Options

- **Hardware Appliance:** I'll cover system architectures, common deployment models, and setup. Then we'll delve into the major benefits and constraints of appliances — including performance, scalability, architecture, and disaster recovery.
- **Software:** I will compare DAM appliances with software architectures, and consider the various deployment models available with each flavor — then cover pros and cons including installation and configuration, flexibility, scalability and performance, and installation/setup.
- **Virtual Appliances:** Virtualization and cloud models have required many security technologies to adapt, including DAM. I will discuss why virtual appliances are necessary, contrast against hardware appliances, and discuss practical considerations.
- **Event Collection and Data Management:** I will briefly discuss how data collection and management affect these architectures, with a focus on questions that come up in competitive situations and tend to confuse buying decisions.

There is no single *best* technology. Every customer has its own preferred way to manage IT systems and security, so the following information is presented to help you make decisions appropriate for *your* environment.

Appliances

It may seem minor, but appliances have a visceral effect on customers. It's the first difference they notice when comparing Database Activity Monitoring products. And frankly, they're impressive — those racks of blinking whirring 1U & 2U machines, neatly racked with substantive presence. Clustered in groups in your data center, with lots of cool lights, logos, and deafening fans. Sometimes called "pizza boxes" by the older IT crowd, these are basic commodity computers with 1 or 2 processors each, memory, redundant power supplies, and a disk drive or two. Inexpensive and fast, appliances make up more than half the world's DAM deployments.

When choosing between solutions, positive first impressions have a huge impact on buying decisions, and this is a big reason appliances have been favorites for years. Everything is self-contained and much of the monitoring complexity can be hidden from view. Basic operation and data storage are self-contained. System sizing — choosing the right processors(s), memory, and disk — are the vendor's concern, so the customers don't have to worry about it or take responsibility (even if they do have to provide all the actual workload data...). To further cement the positive impression, initial deployment is easier for average customers, with much less effort to get up and running.

Advantages

- **Fast and Inexpensive:** The appliance is dedicated to monitoring. You don't need to share resources across multiple applications or worry that another application might impact monitoring, and the platform can be tailored to its function. Hardware is chosen to fit the requirements of the vendor's code; configuration can be tuned to well-understood processor, memory, and disk demands. Stripped-down Linux kernels are often used to avoid unneeded OS features. Commodity hardware can be chosen by the vendor, based purely on cost/performance considerations. On equivalent platforms, appliances perform slightly better than software simply because they have been optimized by the vendor and are unburdened by irrelevancies.
- **Deployment:** The beauty of appliances is how simple they are to deploy. This is their most obvious advantage, although it is most relevant in the short term. Slide it into the rack, connect the cables, power it up, and you get immediate functionality. Most of the sizing and capacity planning is done for you. Much of the basic configuration is in place already, and network monitoring and discovery are available without little or no effort. The box has been tested; in some cases the vendor pre-configures policies, reports, and network settings before shipping. You get to skip a lot of work on each installation. Granted, you only get the basics, and *every installation requires customization*, but the immediate functionality makes a powerful first impression during competitive analysis.
- **Avoid Platform Bias:** "We use HP-UX for all our servers," or "We're an IBM shop," or "We standardized on SQL Server databases." All the hardware and software bundled within the appliance is largely invisible to customers, which helps avoid religious wars over configuration and most compatibility concerns. This makes IT's job easier and avoids conflict with hardware/OS policies. DAM provides a straightforward business function, and so can be evaluated simply on how well it performs that function.

- **Data Security:** The appliance is secured prior to deployment. User and administrative accounts need to be set up, but network interfaces, web interfaces, and data repositories are all pre-configured by the vendor. There are fewer moving parts and less configuration, making appliances more secure than their software counterparts at delivery and simplifying security management.
- **Non-relational Storage:** To handle high database transaction rates, non-relational storage within the appliance is standard. Raw SQL queries from the database are stored in flat files, one query per line. Not only can records be stored more quickly in simple files, but the appliance itself escapes the burden of running a relational database. The tradeoff here is very fast storage at the expense of slower analysis and reporting.

A typical appliance based DAM installation consists of two appliance flavors. The first and most common is small 'node' machines deployed regionally — or within particular segments of a corporate network — and focused on collecting events from 'local' databases. The second flavor of appliance is administration 'servers'; these are much larger and centrally located, and provide event storage as well as command and control interfaces for the nodes. This two-tier hierarchy separates event collection from administrative tasks such as policy management, data management, and reporting. Event processing — analysis of events to detect policy violations — occurs either at the node or server level, depending on the vendor. Each node sends at least all notable events to its upstream server for storage, reporting, and analysis. In some configurations *all* analysis and alerting is performed at the 'server' layer.

Disadvantages

But of course nothing is perfect. Appliance market share is rapidly eroding in favor of software and software-based "virtual appliances". Appliances have been the preferred deployment model for DAM for the better part of the last decade, but perhaps not for much longer. There are several key reasons for this shift.

- **Data Storage:** Appliances use commodity hardware — for data storage that means single or redundant SATA disks. Some compliance initiatives require storing events for a year or more, but most appliances only support up to 90 days of event storage — in practice this is often more like 30-45 days. Most nodes rely heavily on central servers for mid-to-long-term storage of events for reports and forensic analysis. Depending on the size of the infrastructure, these server appliances may run out of capacity and/or performance, requiring multiple servers per deployment. Some server nodes use SAN for event storage, while others are simply incapable of storing 6-12 months of data. Many vendors suggest compatible SIEM or log management systems to handle data storage (and sometimes analysis of 'old' data).
- **Virtualization:** You can't deploy a physical appliance in a virtual network. There's no TAP or SPAN port to plug into. The network's virtual topology often makes it impossible to deploy appliances, *even with software agents* to collect events. Virtualization of networks and servers has undercut appliance deployments, and spawned the 'virtual appliance' options I will discuss later. For now I simply note that a virtual 'appliance' is not a *physical* appliance at all, but an entire software stack extracted from the physical platform and deployed in a virtual machine container, controlled by a Virtual Machine Manager just like any other server or application. This trend is becoming even more prevalent as IT shops adopt cloud services which are inherently virtualized.
- **Scalability:** The single most common customer complaint with the appliance deployment model is cost to scale. With appliances scaling is as simple as adding nodes, but that means there is no cheaper alternative to buying new hardware — upgrading existing systems is not an option, and new appliances are relatively expensive. Network topologies and security systems often prohibit the event collector from remote communication with the appliance, requiring remote installation of the appliance; in essence one appliance *per* database. On the other end of the spectrum we see many event collection appliances required to serve application clusters. Depending on your network design, this can be a minor annoyance or a deal-breaker.

- **Flexibility:** One size does *not* fit all. Monitoring solutions are resource-constrained according to the policies they enforce. For example, every rule requires processing overheard to determine if a query is malicious. Moving from 10 rules to 20 doubles the processing overhead. Behavioral and dynamic monitoring profiles demand significant amounts of memory just to create and maintain the comparison profiles. Compliance projects demand large volumes of storage. How you use DAM changes the resources you need. We all know requirements change, and it's harder to re-provision appliances to support changes in volume of database activity, or in security and compliance requirements.
- **Disaster Recovery:** In case of disaster or other data center outage, appliances must be physically moved to a new data center. Redeployment of software or virtual machines — on whatever hardware is available — is cheaper and faster with software based DAM than with physical hardware, which might need to be purchased and shipped from the vendor. And even standby hardware costs money.

Appliances offer many compelling advantages, but deciding whether appliances are right for you requires careful consideration of your goals and database environment. The takeaway here is that the advantages of appliances are most pronounced **early** in the buying cycle. On the other hand, as we'll see in the next section, software deployment requires more up-front work during installation, configuration, and hardware procurement. Early on, appliances are much easier — their limitations often appear only much later. Keep in mind that the deployment will last much longer than the initial evaluation, and consider the rest of the product lifecycle as you decide how you like it. Ease of deployment *is* very important, but long-term product satisfaction has more to do with the ease of day-to-day operations management so weigh that in your selection process.

Software

Database Activity Monitoring software is deployed differently than DAM appliances. Appliances are *usually* two-tier event collector / manager combinations which divide responsibilities, but software deployments are as varied and diverse as customer environments. Your architecture might consist of stand-alone servers at multiple geographic locations, loosely coupled confederations each performing different types of monitoring, hub and spoke systems, everything on a single database server, all the way up to N-tier enterprise deployments. It's more about how the software is *configured* by the customer to address specific requirements. You can configure software any way you prefer. Most customers use a central management server communicating directly with software agents with collect events — at least initially.

Typically, the management server functions are divided across multiple machines as necessary to increase capacity. Distributing event analysis, storage, management, and reporting across multiple machines enables tuning each machine to its particular task. Large enterprise environments dedicate several servers to analyzing events, with others for relational database storage.

This last aspect — use of a relational database management system — is one of the few major differences between software and hardware (appliance) embodiments, and the focus of the most marketing FUD (Fear, Uncertainty, and Doubt) in this category. Some IT folks consider relational storage a benefit, others a detriment, and some a bit of both, so it's important to understand the tradeoffs. In a nutshell, relational storage requires more resources to house and manage data; but in exchange it provides much better analysis, integration, deployment, and management capabilities. To appreciate the advantages of DAM software deployed on your own hardware you need to understand the differences between deployment architectures, and how relational storage changes both performance and the range of possibilities.

Advantages

- **Flexible Deployment:** Add resources and tune your platforms specifically to your database environment, taking into account the geographic and logical layout of your network. Whether it's thousands of small databases or one very large database, at one location or thousands, it's simply a matter of configuration. Software-based DAM offers a half-dozen different deployment architectures, with variations on each to support different environments. If you choose wrong simply reconfigure or add additional resources, rather than buying new appliances.
- **Scalability & Modular Architecture:** Software DAM scales in two ways: by upgrading hardware, and by adding new systems to distribute the workload. DAM installations scale with processor and memory upgrades, or you can move the installation to a larger machine to process more events. But most customers scale by partitioning the DAM software deployment across multiple servers — generally placing the DAM engine on one machine and the relational database on another. This effectively doubles capacity, and each platform can be tuned for its function. This model scales further with multiple event processing engines on the front end, letting the database handle concurrent

insertions, or by linking multiple DAM installations to the back end database. Each software vendor offers a modular architecture, enabling you to address resource constraints with very good granularity.

- **Relational Storage:** Most appliances use flat files to store event data, while software DAM uses relational storage. Flat files are extraordinarily fast at writing new events to disk, supporting higher data capture rates than software installations on equivalent hardware. But the additional overhead of the relational platform is not *wasted* — it provides concurrency, normalization, indexing, backup, partitioning, data encryption, and other services. Insertion rates are lower, while complex reports and forensic analyses are faster. In practice, software installations can directly handle more data than DAM appliances without resorting to third-party tools.
- **Operations:** Operations plays a surprisingly large role in the decision-making process. Software-based DAM looks and behaves like the existing applications your operations staff already manages. You choose which relational platform to store events on — whether IBM, Oracle, MS SQL Server, MySQL, Derby, or whatever you like. You choose the OS (Linux, HP/UX, Solaris, Windows) and hardware (HP, IBM, Oracle, Dell, etc.) you prefer and already own. There is no need to retrain IT operations staff because management fits within existing processes and systems. You can deploy, tune, and refine the DAM installation as needed, with much greater flexibility to fit your model. Obviously customers who don't want to manage extra software prefer appliances, but they are dependent on vendors or third-party providers for support and tuning — at added cost.
- **Cost:** In practice, enterprise customers realize lower costs with software. Hardware appliances have a set cost per unit, so total cost grows per appliance. Software vendors offer tiered pricing and site licenses — once customers reach a certain database threshold costs don't increase, meaning cost per DAM server and per database decreases. Mid-market customers rarely fully realize this advantage, but ultimately software costs less than appliances for enterprises.
- **Integration:** Theoretically, appliances and software vendors all offer integration with third-party services and tools. All the Database Activity Monitoring deployment choices — software, hardware, and virtual appliances — offer integration with workflow, trouble-ticket, log management, and access control systems. Some also integrate with third-party policy management and reporting services. In practice the software model offers additional integration points and more options. Most of these additional capabilities leverage facilities and procedural interfaces in the underlying relational databases. As a result, software DAM deployments provide more flexibility for business analytics, SIEM, storage, load balancing, and redundancy.

Remember that most of these advantages are not visible during the initial deployment phases or Proof of Concept (PoC). Over the product lifespan, however, these benefits pay off, and are often essential to enterprise customers. While software has many advantages, there are downsides as well, of course.

Disadvantages

- **Time to Install & Configure:** Every software DAM instance must be installed and configured prior to deployment. Hardware appliances come pre-configured, and virtual appliances deploy from snapshots and pre-configured images. You can create installer scripts and images to streamline repetitious installations, but it's still more work to get software up and running.
- **Event Security:** Hardened appliances offer an advantage in security over software. Software installations *can* be as secure as appliance installations but rarely are. Appliances arrive secure, while software needs to be configured to close common weaknesses in the underlying platforms and database software. Software vendors provide guidance and best practices, but given the diversity of customer deployment models they cannot fully automate secure configuration.

- **Patching:** Software, hardware, and virtual appliances *all* need to be patched and updated. Appliances are maintained by the vendors, but with software *you* get to do all the patching.
- **Hardware:** Just because you did not buy an appliance does not mean you don't need to buy hardware. Some organization have hardware spares and can easily provision DAM on inventory they already have, but most need to requisition new stuff. Worse — as I know from practical experience — customers test software deployments (PoC) on old garbage stored in the closet because nobody can use it for real work, while appliance vendors FedEx sleek new boxes with much more oomph. In this kind of rigged comparison appliances inevitably perform *much* better. Just remember that you need to do capacity planning, budgeting, and going up the approval chain for *both* the DAM software and its host hardware. The good news is that most organizations are used to this, but it's still a hassle.

Most DAM vendors see themselves as software vendors — even those whose primary distribution model is bundling their code into hardware. They *do* write software, especially the complex agents that collect events, but the final product is not software. There are significant differences between pure software and appliances — virtual or hardware based — as will become readily apparent when we discuss Virtual Appliances, next.

Virtual Appliances

For Database Activity Monitoring, virtual appliances are a response to the poor fit of hardware appliances in virtualization environments. Management, hardware consolidation, resource and network abstraction, elasticity, and even power savings advantages are nullified in a virtual network. Infrastructure as a Service (IaaS) is also a difficult environment for hardware appliances to fit into. So *every* hardware appliance vendor has packed its application stack into virtual machine images. It's a quick win for them, as very few changes are needed, and they escape the limitations of hardware. A virtual appliance is 'built' and configured like a hardware appliance, but delivered without the hardware. That means all the software — both third-party and vendor-created — of a hardware appliance is instead wrapped in a virtual machine image. This image is run and managed by a Virtual Machine Manager (VMware, Xen, Hyper-V, etc.), but otherwise functions just like a physical appliance — including simpler flat-file data storage as on a physical appliance, rather than fuller-featured relational storage as used by software platforms.

In terms of benefits, virtual appliances are basically the opposite of hardware appliances. Like the inhabitants of the Star Trek mirror universe, hardware and virtual appliances look alike (from a feature standpoint), but act very differently. Sure, virtual appliances share some similarities with their hardware counterparts — including ease of deployment, flat file storage, and lack of hardware dependencies — but many aspects are quite different than both hardware and software DAM platforms.

Advantages

- **Scale:** Taking advantage of the virtual architecture, it's trivial to spin up new appliances to meet demand. As with software, processor, memory, and disk can be upgraded as necessary. Adding new DAM instances is a simple VMM operation, and they can run close to the monitored databases. Multiple instances still collect and process events, and send alerts and event data to central appliances for processing. You still have to deploy software agents on the various database, and manage connections and credentials, of course.
- **Cloud & Virtual Compatibility:** A major issue with hardware appliances is their poor fit in cloud and virtual environments. Virtual instances, on the other hand, can be configured and deployed in virtual networks to both monitor and *block* suspicious activity.
- **Management:** Virtual DAM is managed just like any other virtual machine, using the operational management frameworks and tools the customer has in place. Adding resources to the virtual instance is trivial — unlike most hardware solutions. Patching DAM images is easier, quicker, and less disruptive. And it's easy to move virtual appliances to accommodate changes in virtual network topology.

Disadvantages

- **Performance:** This is in stark contrast to hardware appliance performance. Customers complain about both storage latency and performance. Not running on dedicated hardware has a cost — resources are neither dedicated nor tuned for DAM workloads — as the resources are allocated when the virtual image is started. Event processing performance is on par with software, so generally not a concern. You need to consider storage latency and throughput when planning capacity and setup of VMs. DAM is particularly susceptible to latency because it is designed to support real-time monitoring, so it is important to monitor I/O performance and virtual bottlenecks and adjust accordingly.
- **Elasticity:** Virtual environments are *far* more elastic than DAM applications — virtual DAM appliances are very easy to replicate but you still need to plant for capacity and reconfigure virtual machine for their workload. Additional memory and processing power help, but virtual appliances require configuration to match requirements just like any software.
- **Cost:** Cost may not be either an advantage or a problem, but it is an important consideration when moving from hardware to virtualization. Surprisingly, I find that customers using virtual environments use more and smaller databases, and use more virtual appliances to monitor them. Ultimately, cost depends entirely on the vendor's licensing model. Customers often forget, when moving from hardware to virtual appliances, to renegotiate pricing with the vendor. If you're paying on a per-appliance or per-database model, you don't realize the cost reduction benefit of software.

Customers add virtual appliances to supplement existing hardware deployments — either to fill in capacity or to address virtual networking issues, particularly for enterprise customers. Interestingly, there is no trend to phase either out in favor of the other — customers stick with the hybrid approach.

Event Collection

How you collect events — SQL statements sent to the database, query results, administrative tasks, batch jobs and even database failures — matters. As much attention as we have devoted to deployment architectures and their performance and effectiveness implications, data collection is crucial too. To navigate through the FUD we need to cover a couple competitive topics that get lumped into bake-offs.

One of the most common DAM marketing statements is: "We do not require agents." This is technically correct but completely misleading. Let's look at that claim and other data collection issues in the Appliance vs. Software debate.

Impact

- **Yes, We Have No Agents:** No Database Activity Monitoring solution *requires* an agent. You'll hear this from all the vendors to address the competitive 'poison pill' left by a previous vendor. All but one DAM product can collect SQL and events *without* an agent. But the statement "We don't require an agent" is pure marketing. In practice *all* DAM products — software, hardware, and virtual — use agents. They do this because agents, of one form or another, are the only reliable way to make sure you get *all* important events. It's how you get the complete coverage necessary for security and compliance. Nobody serious about compliance or security skips installing an agent on target databases.
- **No Database Impact:** Since you will be using an agent, there will be an impact on database performance. The agent may collect SQL from the network stack by embedding into the OS; or by scanning memory; or by collecting trace, audit, or transaction logs. Don't believe the "no impact" clams. If a vendor says this, they're referring to inadequate *agent-less* data collection. That, and increased administrative responsibilities to maintain the agent.
- **Network Capture:** The alternative to using agents is network data capture. Sure, vendors offer pure network traffic collection to monitor for external threats, but that model *fails to collect critical events* on the database platform itself. If you don't collect all the important events you cannot reliably enforce all security policies. Don't get me wrong — network capture is great for detecting a subset of security specific events, and it's even preferable for non-critical databases, but network scanning fails to satisfy compliance requirements. But agent-less deployments are still common because they detect many security threats without needing to manage agents on the target database.
- **Complete SQL Activity:** DAM focuses on collection of database events. Agents that collect from the network stack outside the database, or directly from the network, focus on raw unprocessed SQL statements in transit before they get to the database. For many customers just the SQL statements are enough, but more often the *result* of the SQL statement is just as important. The number of rows returned, or whether the query failed, is essential information. Many network collectors do a good job of query collection, but poor result collection. In some cases they capture only the result code, unreliably — I have seen capture rates as low as 30% in live customer environments. For operations management and forensic security audits this is unacceptable, so you need to verify coverage.
- **Database Audit vs. Activity Audit:** If the power goes out and 100k database transactions get rolled back, that's an important event for compliance reporting. It's not SQL queries issued to the database, but it certainly is database

activity — the contents of the database changed. This sort of event makes it important to differentiate between an activity audit and a database audit. If your agent collects data *from outside the database*, you are auditing *activity*. If you collect data from *inside the database,* you are auditing the *database* — it's that simple. And this is a very important distinction for compliance, where you may need to know database *state*. It is *considerably more difficult* to collect from database memory, traces, transaction logs, and audit logs. Using these data sources has more performance impact — anywhere from *a bit* to *much* more than activity auditing, depending upon the database and the agent configuration. Worse, database auditing doesn't always pick up the raw SQL statements. But these data sources are important because they provide insight into the state of the database and *transactions* — multiple statements logically grouped together — that activity monitoring handles less well. Very few customers combine both methods to monitor any single database — one method or the other is usually sufficient.

Conclusion

There is no single 'best' deployment model for Database Activity Monitoring — each option imposes its own tradeoffs. But make no mistake — the deployment model is critical. Hardware, software, or virtual appliance — each has its own advantages and disadvantages. What works best for any customer depends on the specific requirements. And just like vendors, customers (including you) have biases to consider alongside security and compliance requirements.

I can't stress enough that long-term happiness with DAM has little to do with the proof of concept hurdles. Products that sail through product bake-offs without a hitch may later turn out to cost 4 times as much when implemented across the entire organization. I have seen products scale to meet customer requirements, but require a dedicated full-time administrator to manage. I have seen appliances deployed and ripped out 9 months later, because the data center shifted to virtualization. I have watched IT organizations buy based on the "No Agent Required" promise, only to lean later that they really needed agents on every database.

You will have security and compliance requirement checklists you need to address, but the functional capability differences between vendors are shrinking. As a result vendors have shifted the discussion to other areas, with deployment the current favorite. Changing trends in foundation IT technologies — hardware to virtualization, and virtualization to cloud services — alter the base assumptions. They change how we install and manage systems, and how we go about monitoring databases. You need to understand the pros and cons of each model to decide which will work best for you, day in and day out, as your organization changes.

Who We Are

About the Author

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 within 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 security 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. Adrian can be reached at alane (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.

We provide services in four main areas:

- Publishing and speaking: Including independent objective white papers, webcasts, and in-person presentations.
- Strategic consulting for end users: Including product selection assistance, technology and architecture strategy, education, security management evaluations, and risk assessments.
- Strategic consulting for vendors: Including market and product analysis and strategy, technology guidance, product evaluations, and merger and acquisition assessments.
- Investor consulting: Technical due diligence including product and market evaluations, available in conjunction with deep product assessments with our research partners.

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.

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