Security Analytics with Big Data

Version 1.1
Released: February 19, 2014

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.

Contributors

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

Danny Banks
Alex C
Javier Jarava
Anton Chuvakin
Matthew Gardiner
Ulf Mattsson

Licensed by Vendor

Securosis is an information security research and advisory firm dedicated to transparency, objectivity, and quality. We are totally obsessed with improving the practice of information security. Our job is to save you money and help you do your job better and faster by helping you cut through the noise and providing clear, actionable, pragmatic advice on securing your organization.

For more information visit securosis.com.

Copyright

Table of Contents

Introduction

Big data is touted as a 'transformative' technology for security event analysis — promised to detect threats in the ever-increasing volume of event data generated from in-house, mobile, and cloud-based services. We hear big data will do more, do it better, and cost less. IT and security personnel, having seen this type of hyperbole many times, are justifiably skeptical: we have been promised rainbows and flying unicorns before. The combination of industry hype, vendor positioning, and general confusion in the press about big data 'is' makes seasoned security folks all the more wary. And if we've learned anything when it comes to security products, a healthy dose of skepticism is a good thing. Most IT and security practitioners do not yet fully understand big data, nor how to apply it to their own requirements, so they are unable to weed through all the hype so the remain reticent to embrace the technology.

But here's the thing: Big data clusters have been in production use for the last decade, and have proven their worth for both search capabilities at companies like Google and traditional data mining. Big data inherently offers advantages for scaling and advanced analytics, both of which can be applied to threat detection.

This is not just theory and speculation: big data is currently being employed to detect security threats, address scalability requirements for IT event management, and even help gauge the effectiveness of other security investments. It actually works! Big data natively addresses ever-increasing event volume and the rate at which we need to analyze new events. There is no question that it holds promise for security intelligence, both in the varied ways it can parse information and by its native capabilities to sift proverbial needles from monstrous haystacks. Cloud and mobile services are forcing us to reexamine how we manage security and to scale across broader sets of systems and events — neither of which fits well into the structured data repositories on which most organizations rely. The volume, velocity, and variety of data tax existing systems beyond their limits — but that's only half the story. The tools we are using to detect attacks are not particularly effective and they are less agile than our attackers'.

This research paper will describe what big data is and apply its essential characteristics to data mining for security events. We will offer a clear and *unique* definition of big data, and explain how it helps overcome current technical barriers. We propose a pragmatic way for customers to leverage big data, enabling them to select a solution strategically. The paper discusses use cases driving the search for new solutions, how big data maps to these needs, and delves into what it takes to deliver security analytics. We have seen that many readers don't fully understand what big data is, so we will start with a definition which should will give you confidence that you can recognize big data and understand the basics of how it works. We will focus on areas where big data excels, and discuss the challenges in developing big data systems and security analytics engines.

SIEM is not Big Data

When we started this research our goal was provide a clear understanding of how big data can be leveraged to advance security analytics beyond what is currently offered. But that original goal evolved as we asked customers what specific questions they wanted us to address. Almost universally they asked, "Help us understand how to leverage our current investments in SIEM and make that better," and explained, "We don't need a totally new system, we need what we have to work." Still others considered SIEM their compliance platform, but when it came to threat analysis and support for security operations, many want a big data platform to specialize in analysis and analytics. What they needed to know is

how to get big data and SIEM to cooperate, understanding they are different platforms. Our goal is to educate readers on a) what big data is, b) how it can improve security analytics, and c) how it will — or won't — integrate with SIEM. That's not to say that SIEM vendors will provide big data distributions as part of their solution, rather most will architect big data techniques into their platforms to deliver similar value over time. Some already provide big data-like clustered solutions that scale *very well*, but don't offer the full integration capabilities of big data.

We understand there will be some controversy, as a handful of respondents pointed out. We do not call big data SIEM 2.0, or bash SIEM for its failings — neither is useful. But we find that the *contrast* between SIEM and big data is instructive for people who don't have experience with big data. SIEM is a mature technology, and is well understood by security and IT operations teams, but it was designed and built for the security challenges of 8-10 years ago. It is not only a case of scaling up to a much larger data set; we also collect more types of data, each requiring new and different analysis techniques to detect advanced attacks. All that data slows down SIEM and log management systems, but of course you are under the gun to identify attacks *faster*.

We will endeavor to balance our discussion between buy vs. build, looking at the challenges of each. Where helpful for demystifying big data, we will contrast traditional data management systems, and highlight areas where big data excels in comparison. We will also discuss some changes in operational processes and mindset necessary to take full advantage of this new technology.

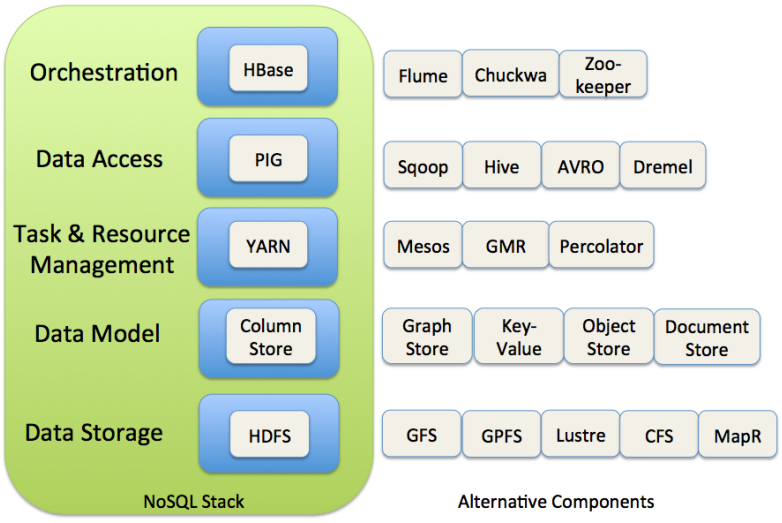
Defining Big Data

Big data is not really about data; it's about tools that manage and derive value from data. The media often equates big data with 'a lot' of data, meaning many terabytes — or even petabytes — of data. But having lots of data is not what the trend is about about. Even Wikipedia's big data page falls into this same trap, equating big data with lots of data, stating "big data tools are being developed to handle various aspects of large quantities of data", but we've had multi-terabyte databases for a decade or more. To their credit, Wikipedia's description (now) includes the principal challenges big data systems must address: increased Volume (quantity of data), Velocity (rate of data accumulation), and Variety (different types of data) — also called the 3Vs. But while this get's the reader one step closer, the **Wikipedia definition fails to capture the essence of big data** as it does not capture the facets of big data *technology* that have made it a disruptive force in IT.

But this is somewhat understandable as defining big data is not easy. While we historically have had no trouble defining relational databases based on common elements such as table-based data storage, relational constructs, and transactional consistency — big data is not so simple. The variability of big data clusters is part of what make them so attractive. Let's approach a definition as a developer would, and look at how we would assemble a big data solution to get a better idea of how these platforms work.

Essential Characteristics of Big Data

The current poster child for big data is Apache Hadoop, an open source platform based on Google BigTable. Hadoop is not the first big data database, but it is the most commonly used. A Hadoop installation is built as a clustered set of commodity hardware, with each node providing storage and processing capabilities. Hadoop provides tools for data storage, data organization, query management, cluster management, and client management.



It is helpful to think of the Hadoop framework as a 'stack' like the LAMP stack. The Hadoop components are normally grouped together but you can replace each component or add new ones as desired. Some clusters add optional data access services such as Sqoop and Hive. Lustre, GFS, and GPFS, can be swapped in as the storage layer. Or you can extend HDFS functionality with tools like Scribe. You can select or design a big data architecture specifically to support columnar, graph, document,

XML, or multidimensional data. This modular approach enables customization and extension to satisfy specific customer needs.

But that is still not a definition. Nor is Hadoop the only player. Users can choose Cassandra, Couch, MongoDB, Splunk or RIAK instead — or investigate 150 alternatives at this time. Each is different — focused on its own particular computational problem area, replicating data across the cluster in its own way, with its own storage and query models, etc. One common thread is that every big data system is based on a 'NoSQL' (non-relational) database; they also embrace many *non*-relational technologies to improve scalability and performance. Unlike relational databases — defined by use of relational keys, table storage, and various other common traits — there is no such commonality among NoSQL platforms. Each layer of a big data environment can be radically different so there is much less common functionality than between RDBMS.

We have seen this difficulty defining platforms before — the term "Cloud Computing" used to be similarly meaningless, but we came to grips with the many different cloud service and consumption models. We lacked a good definition until NIST defined cloud computing based on a series of essential characteristics, a clever approach to defining a complex and highly variable subject. So we took a similar approach for big data, defining it as a framework of utilities and characteristics common to all NoSQL platforms.

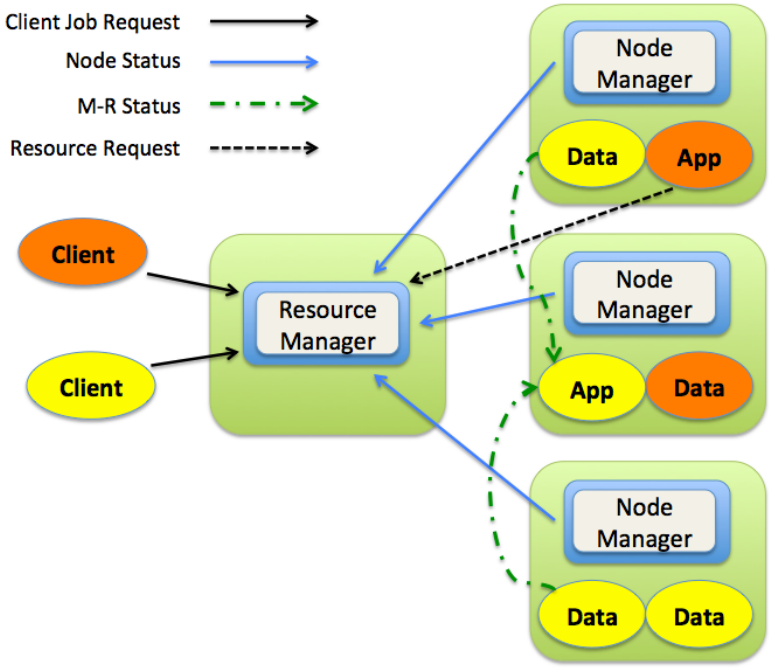
- **Very** large data sets (Volume)
- Extremely fast insertion (Velocity)
- Multiple data types (Variety)
- Clustered deployments
- Providing complex data analysis capabilities (MapReduce or equivalent)
- Distributed and redundant data storage
- Distributed parallel processing
- Modular design
- Inexpensive
- Hardware agnostic
- Easy to use (relatively)
- Available (commercial or open source)
- Extensible — designers can augment or alter functions

As you can see, there are more essential characteristics to big data than just the 3Vs. Additionally we could include variability for data management, cost reduction, more extensive analytics than possible with SQL-style queries, and customization (including a modular approach to orchestration, access control, task management, and query processing). However, I've excluded them from the essential characteristics as some relational platforms make these same claims, and the above list provides ample differentiation.This broader collection of characteristics captures the big data value proposition, and offers a better understanding of what big data is and how it behaves.

What does it look like?

The next page shows a typical big data cluster with multiple nodes cooperating to manage data and process queries. A central resource manager, sometimes called "the name node", manages the cluster and client connections. Clients communicate directly with the name node with job requests and individual data nodes as necessary for query results.

This diagram shows the critical components and the communication between them. The resource manager handles cluster administration, and each node in the cluster has a local node manager to help coordinate data management on the local node as well as between nodes. But keep in mind that our simplified diagram reflects just three data nodes and two client applications — a big data cluster might easily comprise 500 nodes hosting 30 applications. More nodes enable faster data insertion and parallel query processing improves responsiveness substantially. 500 nodes would be overkill to support your SIEM installation, but it's nice to know you can scale up if you need it.



Why Companies Adopt Big Data

Thinking of big data systems as simply large databases, or even limiting one's definition to the 3Vs, is reminiscent of The Blind Men and the Elephant, where each man only perceives one facet of the whole and none actually recognize the elephant.

The popularity of big data is largely due to its incredibly cheap analytics capabilities. The big data revolution has been driven by three simple evolutionary changes in the market: inexpensive commodity computing resources, availability of a boatload of interesting data to analyze, and — most importantly — *virtually free analytics tools*. Together they provide value which has spurred widespread demand. Once organizations see what big data can do for marketing and sales data, they wonder what it can do for other computational challenges such as threat and fraud detection.

The big data revolution did not occur because companies of all sizes suddenly stumbled over millions of dollars earmarked to buy "big iron", but because they are rapidly learning to perform advanced analytics for pennies. We have

been able to build systems which can process and store vast quantities of data for a couple decades, but they required multi-million-dollar investments to even get started. These large and expensive systems were scare, data was much more costly, and management and programming personnel were incredibly rare (and thus expensive).

Don't Be 'Ha-duped'

It is helpful to keep these three drivers in mind when considering what big data is and is not. We could spend another 20 pages defining big data and describing Hadoop — the big data poster child —in detail, but we have enough to make meaningful comparisons between different solutions. If an underlying data management architecture does not support distribution of queries across nodes it will be less able to deliver timely information than a true big data system. If a "big data" infrastructure is proprietary, that's OK as long as it meets a sufficient sub-set of essential criteria we mentioned above. The downside is it will be harder to leverage open source or commodity commercial tools to extend its capabilities with proprietary infrastructure, or to find people who can implement advanced policies and build security analytics.

It's critical to ascertain if your vendor is really using big data or is it the same thing they have been selling all along? You need to understand what big data is so you can tell whether a vendor's BD offering is valuable or snake oil. Don't be 'Ha-duped'! Some vendors are deliberately sloppy with terminology, with "big data" offerings that are not really big data at all. It might be a relational data store with a "Big Data" sticker, or a proprietary flat file data storage format without any of the features that make big data platforms powerful.

Big data means better scalability, better analysis, and faster results at lower cost. If the essential characteristics are not present, it's likely you are missing out on one or more core advantages.

Use Cases

Why use big data for security analytics? Aside from press hype, what motivates customers to look for new solutions? On the other side of the coin, why are vendors altering their products to use — or at least integrate with — big data? Customers cite performance and scalability as drivers — particularly for security event analysis. This research project was originally sketched out as a broad examination of big data's potential for security analytics. The customers we speak with don't care about generalities — they need to solve existing problems, specifically around installed SIEM and log management systems. APT, Malware, fraud detection and the like are beyond many existing systems detection capabilities. And more firms are leveraging the same data to perform business analytics and operational risk measurements. The use cases we describe here refocused on the underlying challenges of getting the right data and the right analytics in order to address these problems, and this is how big data is causing disruption in the SIEM marketplace.

SIEM, log management, and other event-centric monitoring systems that struggle under evolving requirements. Big data is quite the opposite: It's inherent strengths line up particularly well with SIEM's deficiencies in the areas of scalability, analysis speed, and rapid data insertion. And given the potential for greater analysis capabilities, big data offers great promise for keeping pace with exploding event data volumes and taking better advantage of it.

Let's look at how big data tackles those issues.

More (Types of) Data

The problem we heard most often was "We need to analyze more *types* of data to get better analysis". The need to include more data types, beyond traditional netflow and `syslog` event streams, is to support derivation of actionable information from the sea of data. Threat intelligence is much more than a simple signature, and detection is much more involved than reviewing a single event. Communications data such as Twitter streams, blog comments, voice, and other rich data sources are unstructured and require different parsing algorithms to interpret. Netflow and `syslog` data is highly structured, with each element defined by its location within a record. Blog comments, phishing emails, botnet command and control (C&C), and malicious files? Not so much. The problems with accommodating more types of data are scalability and usability. First, more data types means more data, and existing systems often can't handle any more. Increasing capacity of fully taxed systems often requires costly add-ons. Rolling out additional data collectors and servers to process their output takes months, and the cost in IT time can be prohibitive. That all assumes the SIEM architecture *can* scale up to greater volumes of data coming in faster. Additionally, many of these systems simply cannot handle alternative data types — either they normalize the data in a way that strips much of its value, or they lack the tools to analyze alternate (raw) data types. Most systems have evolved to include configuration management and identity information, but not Twitter feeds or diverse threat intelligence. In the face of evolving attack profiles, flexibility to capture and dig into any data type is now a key requirement.

Addressing "What Else?"

We have seen steady advances in aggregation, correlation, dashboards, and data enrichment to help security folks identity security threats faster. But these iterative advancements have not kept pace with the volume of security data to be parsed or the diversity of attack signatures. Overall situational awareness has not improved and the signal-to-noise ratio has gotten worse rather than better. A big part of the problem is sanitized data; shrunken, stripped and compresses to optimize storage and ease of processing. What is needed is more data, of more types, logically linked together to optimize finding bad behavior, as opposed to saving storage and processing costs. Some view an alert of suspicious activity as the end goal, when it's really just the initial loose thread where you begin your investigation.The tools to both give you clues on where to look, and help you uncover what you are looking for are lacking. In fact, you're often looking for *something* without knowing exactly what it is you are looking for, so you need easy and flexible search capabilities.

Addressing the question of "what else is there to see" requires more powerful analytics and data mining tools to support what you do today. That means keeping raw data along with the meta-data. That means better enrichment and cross referencing. It means keeping, or adding, context. It means more types of data, and having the system parse both machine events long with human generated — and human readable — data. It means pre-processing complex views so the cluster does the heavy lifting for you. What it really comes down to is a merger of more data, of more types, with better referential capabilities that allows us to drill into data in novel ways. We may still call it 'drill-down', but we turn data upside-down, inside-out and slice it up as needed; whatever provides the clearest possible *views* into a sea of events.

Architectural Limitations

Some customers attribute their performance issues — especially lagging threat analyses — to SIEM architecture and process. It takes time to gather data, move it to a central location, normalize, correlate, and then enrich. This generally makes near-real-time analysis a fantasy. Queries run on centralized event servers and often take minutes to complete, while compliance reports generally take hours. Some users report that their data volumes stress their systems and relational queries take too long to complete. The classical solution is centralized management and reporting with distributed processing, but administration of multiple servers is challenging in many existing implementations.

Disruptive Innovation

A handful of large enterprises have a broad set of issues that encompass everything listed above and more. "We have more customer data to manage and more regulations to address, and we are adding more applications that use this data. At the same time we are moving some systems to the cloud and providing mobile access. We manage employee corporate and personal identities across these systems. And we need to detect misuse while all these changes are underway. Is this even possible?" This is another case for scalability, enhanced analysis, and high-speed data insertion. In addition, cloud services that offer event data produce variable event log formats. Employees using multiple personae on different devices further complicate correlation. Cloud services, mobile devices, and changes in identity management all make SIEM more difficult.

Each use case represents a different set of issues. It is tempting to lump all these issues into a single use case. More data, better data, better analysis, and faster analysis — you can sum them up by saying "do more with more". But the symptoms driving the big data discussion differ for every customer, so appropriate solutions vary.

How Big Data Advances Security Analytics

So why are we looking at big data, and what problems can we expect it to solve that we couldn't before? Most SIEM platforms struggle to keep up with emerging needs for two reasons. The first is that threat data does not come neatly packaged from traditional sources, such as `syslog` and netflow feeds. There are many different types of data, data feeds, documents, and communications protocols that contain diverse clues to data breaches or ongoing attacks. Users demand analysis of a broader data set, in hope of detecting advanced attacks. The second issue is that many types of analysis, correlation, and enrichment are computationally demanding. As with traditional multi-dimensional data analysis platforms, crunching the data takes horsepower. More data is being generated; add more types of data we want, and multiply that by additional analyses, and we get a giant gap between what *we need to do* and what *we can presently do*.

In the Defining Big Data section we described what big data *is.* Now let's talk about how NoSQL database architectures address several of the SIEM pain points. In particular, finding security events we don't know how to describe.

Analysis: Looking at More

Two of the most serious problems with current SIEM solutions are that they struggle with the amount of data to be managed, and they cannot deal with the "data velocity" of near-real-time events. Additionally, they need to accept and parse new and diverse data types to support new types of analysis. There are many different types of event data, any of which can contain clues to security threats. *It's our belief that better security will come from linking more types of event data together in more meaningful ways!* To better understand what we mean, here are some new types of data:

- **Human-readable data:** There is a great deal of data which humans can process easily but which is much more difficult for machines, including blog comments, audio and Twitter feeds. Tweets, discussion forums, Facebook posts, and other types of social media are all valuable for threat intelligence. Some attacks are coordinated in discussion fora so companies want to monitor them for warnings of possible or imminent attacks, and perhaps even details of the threats. Some botnet command and control (C&C) communications occur through social media, so there is potential to detect infected machines through this traffic.

- **Telemetry feeds:** Cell phone geolocation, lists of sites serving malware, mobile device IDs, HR feeds of employee status, Skype (at least for the government) and dozens of other real-time data feeds indicate changes in status, behavior, and risk profiles. Some of these feeds are analyzed as the stream of events is captured, while others are collected and analyzed for new behaviors. There are many different use cases but security practitioners, observing how effectively retail organizations predict customer buying behavior, seek the same insight into threats.

- **Financial data:** We were surprised to learn how many customers use financial data purchased from third parties to help detect fraud. The use cases centered around SIEM for external attacks against web services, but they organizations also analyze financial and buying history to predict misuse and account compromise.

- **Contextual data:** This is anything that makes *other* data more meaningful. Contextual data can indicate automated processes rather than human behavior — an impossibly rapid series of web requests, for example, probably indicates a bot rather than a human customer. Contextual data also includes risk scores generated by arbitrary analysis of metadata and detection of odd or inappropriate series of actions. Some is simply collected from a raw event source, while other data is derived through internal analysis. As we improve our understanding of where to look for attack and breach clues we leverage new data sources and examine them in new ways. SIEM generates some contextual data today, but collection of a broader variety of data enables better analysis and enrichment.

- **Identity and personas:** Today many SIEMs link with directory services to identify users. The goal is to link a human user to their account name. With cloud services, mobile devices, distributed identity stores, identity certificates, and two-factor identity schemes, it has become much harder to link human beings to account activity. As authentication and authorization facilities become more complex, SIEM must connect to and analyze more and different identity stores and logs.

- **Metadata:** Metadata describes other data. It Is not what we call 'content' proper, but instead describes attributes of other data. Sometimes it is helpful to look at the forrest through the trees, and detecting threats is a matter of looking at the big picture. Some threat analysis and botnet detection works from metadata rather than directly from actions.

- **Network data:** Some of you are saying "What? I thought all SIEMs looked at network flow data!" Most do but not all. Some collect and alert on specific known threats, but only a tiny portion of the raw data comes down the wire. Cheap storage makes it feasible to store more network events and perform behavioral computation on general network trends, service usage. While we used to rely upon aggregation and normalization, we'll now see more derived (*i.e.* pre-computed) views of network traffic; for those of you familiar with more traditional data mining this is one dimension of a multi-dimensional cube, which will be used with other derived and raw data to detect threats.

Each of these examples demonstrates what will be possible in the **short** term. In the **long** term we may be able to record any and all useful or interesting data. If we can link in data sets that provide different views or help us make better decisions, we will.

We already collect many of these data types, but we have been missing the infrastructure to analyze them meaningfully.

Analysis: Doing It Better

One limitation of many SIEM platforms is their dependence on relational databases. Many SIEM platforms still rely upon relational database engines to support part of the analysis process, but have stripped away relational constructs that limit insertion performance. The fundamental relational database **architecture** was designed and optimized for relational queries across a set of tables. Big data platforms implement entirely different storage structures, with different data access constructs which can be bolted together using filtering, tagging, and indexing. A better way to think about this is you're not tied to things you don't need; you can upgrade components that specifically fit your requirements. The cluster can be tailored to suit the specific tasks, such as very fast lookups, or parsing complex data, or linking together multi-dimensional data cubes. We can bolt on tools like Pig, Piql, Mahout, Crunch, AVRO, Hive, Dremel, which provide many more options for data management. Nor are we limited to a single query/access model, as some encourage different *non-SQL* languages. Big data can support very complex comparisons, bundled in different programing languages, and optimized for different types of analyses. This enables better and faster analysis, as well as new analyses that are simply impossible in a pure relational database environment running SQL style comparisons. Many NoSQL distributions support SQL queries, and even more offer SQL-like syntax, but big data allows a much broader range of query options.

Analysis: Doing It Faster

One essential characteristic of big data is the way its architecture scales up to very large data sets — clustering breaks up analysis and data management into smaller, more manageable chunks. Commodity hardware and open source software make it cost effective. But in one key way big data is exactly the opposite of a relational platform. Relational databases focus on a central, confined data model — all data is brought to a central powerful location for management. Big data, in contrast, scatters data across many different servers. Big data systems leverage whatever computing power is available, preferably near the data.

This is critical because requests are no longer bottlenecked by a single large server — instead they can be shared across many — possibly hundreds — of smaller servers. But unlike RAC and GRID architectures, we have several advantages including self organizing data, better parallel processing and — given redundant copies of data in the cluster — we can 'move' computation to the data rather than move data to a compute node. The most common architecture is MapReduce. In this model a query is distributed (mapped) to many different nodes — possibly thousands. Each node examines only its own small subset of data, maps it against the specified query, and returns its own matches. The results from all nodes are filtered and de-duplicated, yielding the 'reduced' result. There are other common architectures, such as columnar storage, which yields incredibly fast lookup and insertion at some expense to flexibility of analysis. Regardless of the storage model, the cluster coordinates all work among the nodes, which enables hundreds of nodes to solve smaller problems in parallel.

Another way to look at this is we have the processing power to answer some basic questions, for example "What is normal?" Just as traditional data mining systems do so well, we can pre-compute views into the the data that show us what a users behavioral profile is, or what network traffic looks like, or link common actions that indicate malware. Then we can compare these pre-computed profiles with incoming events to detect outliers and malicious behavior. Just as with multi-dimensional data models, we can pre-compute views that form a lens to view events streams with more clarity. Big data makes it feasible for us to regularly compute and update these profiles.

Comments on SQL

When you think about relational databases, the first thing that comes to mind is the SQL query language. It's the standard interface for *every* relational database of the last two decades, and has been so ubiquitous that every IT admin and programmer knows how to write basic SQL queries. But while SQL is closely associated with relational platforms it's important to understand that SQL is just a language used to describe a search. In fact — ironic though it may be —SQL query capabilities are often bolted onto NoSQL platforms. The queries are mapped to the NoSQL infrastructure on the users behalf.

This is important for two reasons. First, it provides a known interface to NoSQL platforms. You don't need to learn a programming language or understand the intricacies of big data to run a search. This benefits that non-programmers as they can craft queries without having to also be big data experts. Second, you can leverage your library of existing queries with little modification. It's a huge savings in time when you do not need to rewrite and retest all searches as you introduce big data. We'll thoroughly discuss these benefits in the upcoming section on deployment.

That said, the majority of the big data movement is moving away from SQL queries. While the overwhelming majority of SIEM vendors — and users — are sticking with SQL, less than 20% non-SIEM applications were using the SQL query language. Within many NoSQL environments, SQL queries limit the power and flexibility of the platform. The language cannot capture nuances, nor leverage many advanced features of big data clusters. SQL is a trusty hammer, but we are running into more and more problems which don't look like nails.

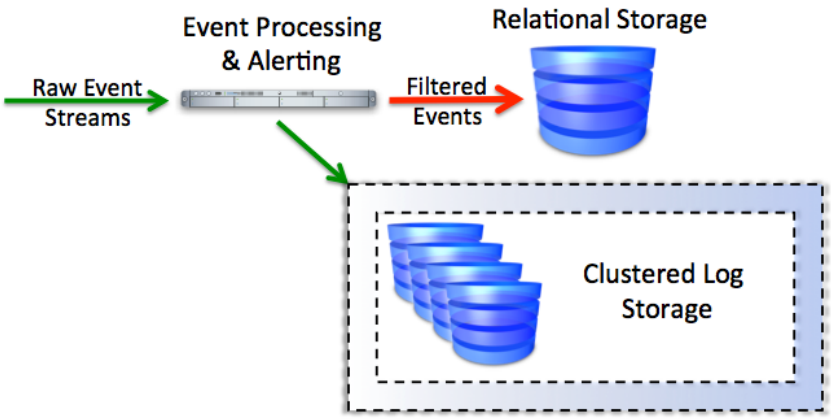
Integration

Some of our first customer conversations about big data and SIEM centered on how to integrate the two platforms. Several customers wanted to know how to pull data from existing log management and analytics systems into big data platforms. Most were told by their vendors that big data was to be integrated into their existing solutions; they wanted to know what the integration would look like and how it would affect operations. You probably won't be integrating the two platforms yourself, but you will need to live with your vendor's design choices. The benefit you derive depends on those choices.

There are three basic models for integrating big data with SIEM:

Log Management Container

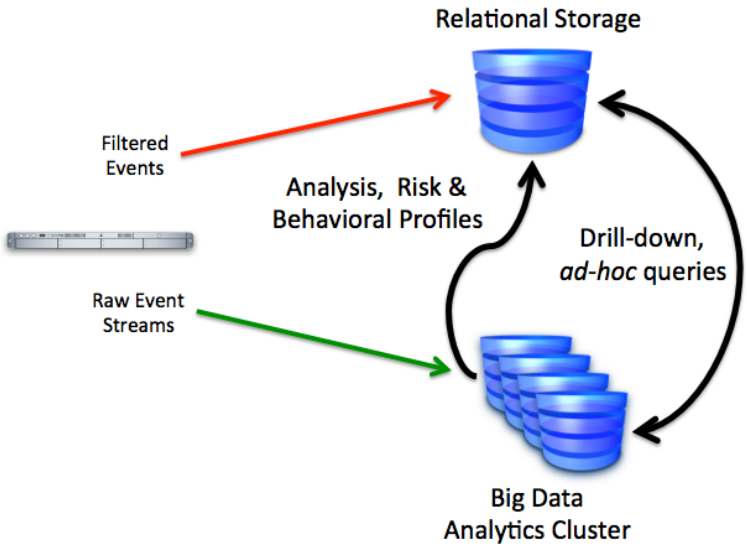
Some vendors choose to integrate big data while keeping their basic architecture intact: a semi-relational or flat file system which supports SIEM functions, fronting a big data cluster which handles log management. We say 'semi-relational' because it is typically a relational platform such as Oracle or SQL Server, stripped of many relational constructs to improve data insertion rates. SIEM's event processing and near real-time alerting remain unchanged: event streams are processed as they arrive and a subset of events and profile information are stored within a relational database or proprietary flat files. Data stored within the big data cluster may be correlated and enriched, but normalization is only performed at the SIEM layer. Raw events are streamed to the big data cluster for long-term storage, possibly compressed. SIEM functions may be supported by limited queries to reference specific data points within the big data archive. Big data is used to scale event storage and accommodate events — regardless of type or format. This is a security data warehouse.



Peer-to-Peer

Like the example above, in this scenario real-time analysis is performed on the incoming event stream, and basic analysis performed in a semi-relational or flat file database. The difference is functional rather than architectural: the two databases are truly peers — **each provides analysis capability** (unlike the model above). Big data acts as the analytics engine, working on structured and unstructured data, with the security 'layer' on top. The big data cluster periodically re-

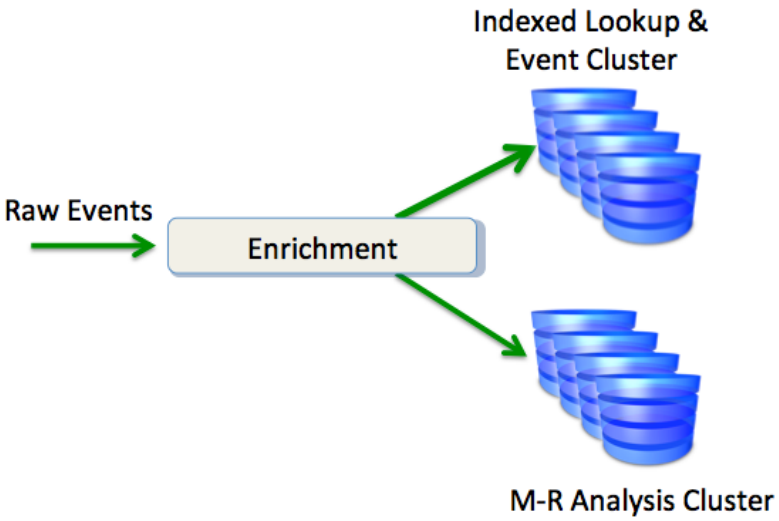
calculates behavioral profiles and risk scores, and shares these with SIEM's real-time analysis component. It also processes complex activity chains and multiple events tied to specific locations, users, or applications may that indicate malicious behavior. The big data cluster does a lot of heavy lifting to mine events, and shares these profiles with SIEM to improve policy enforcement. The big data cluster also provides a direct view for Security Operations Centers (SOC) to run *ad hoc* queries across t complete set of events to look for outliers and anomalous activity.



Full Integration

The next option is to leverage *only* big data for event analysis and long-term log storage. The cluster will be constructed from components that support the specific types of analysis needed, without any legacy baggage. In fact, we expect that clusters will have both dedicated columnar look-up and M-R analysis engines side-by-side, each tuned to maximized performance for their intended task. Even at this time, compute and storage costs are so low that it makes more sense to run parallel databases for specific searches as it saves time and provides faster analysis for threat detection.

You will notice that both peer-to-peer and log management oriented systems use *two databases*; one relational and one big data. There is really no good reason to maintain a relational database alongside a big data cluster — in fact most relational features are irrelevant to security analytics, so they are either stripped out for performance reasons when and where possible. In the future we expect the use of relational databases to entirely give way to NoSQL as they are more suitable to the required tasks. Big data clusters *can be* designed to perform ultra-fast queries, or efficient large scale analysis, or both types of queries on a single data set.



Even if the relational component goes away, we expect *more than one* big data clusters to be the norm because there are different types of jobs to be performed. One cluster might run fully indexed SQL queries for fast retrieval while another runs MapReduce queries to find statistical outliers. In terms of implementation, you could choose Cassandra for its index capabilities and native compression, and Hadoop for MapReduce and large-scale storage. The graphic to the right shows this possibility. It is also possible to have *one* data cluster with multiple query engines running against the same data set. The choice is up to your SIEM vendor depending upon how they want to architect the solution and the performance goals they wish to achieve. During our discussions several vendors were moving to big data only, with a couple employing multiple big data databases in parallel.

Standalone Option

Those of you keeping score at home will notice I am throwing in a fourth option: the standalone or non-integration model. Some of our readers are not actually interested in SIEM at all — they just want to collect security events and run their own reports and analysis. It is perfectly feasible to build a standalone big data cluster for security and event analytics. Choose a platform optimized for your queries (fast, efficient, or both), the types of data to mine, and developer comfort. But understand that you will need to build a lot yourself. A wide variety of excellent tools and logging utilities are available as open source or shareware, but you will be responsible for design, organization, and writing your own analytics. Starting from scratch is not inherently bad but all development (tools, queries, reports, etc.) will fall to your team.

Home Grown Big Data

Our discussion has focused on the most common big data platforms (Hadoop, Cassandra, MongoDB, RIAK, etc) mostly because they help us define what big data is, and point you to specific references that illustrate what's possible. If you're building your own analytics cluster from scratch, you'll likely use one of these. But keep in mind that SIEM vendors are constructing their own versions to suit their needs. Most do *not* leverage these standard platforms, rather they built clusters and query engines that are tuned to the data flows, analytics and orchestration they need. Some of you will be worried about a lack of standards, but openness and flexibility are what's important here.

Integration bias

Think of these three models presented above as evolutionary stages, with most SIEM vendors moving from log management to peer-to-peer. Keep these models in mind and figure out what your vendor is doing because they pretty much all claim more big data integration than they really provide; those running on proprietary file layouts tend to claim they have arrived at the promised land of perfect and effortless data storage. But vendors have a good reason to fudge: you.

For years SIEM vendors have been hearing about their shoddy integration between SIM, SEM, and log management. Their failures to integrate management consoles, policies, and other operational tasks have always been a sore spot with customers — creating a competitive divide for years. Having repeatedly been beaten-up for this, vendors they will claim 'full integration' with big data, but what this means is not clear. You'll need to dig in to their architectural model to understand what 'full integration' means, but don't get too hung up on this. A loose coupling of the data processing capabilities is fine. What is important is how the management console works to unify these pieces, and how easy you can write policies without needing to understand the underlying details. Tight integration in a clustered environment is not your goal. *What you are looking for is central management of distributed data processing!*

Deployment Considerations

Install any big data cluster and you will notice that the documentation focuses on how to get up and running quickly and all the wonderful things you can do with the platform. The issues you really want to consider are left unsaid. That's because many of the day to day deployment and management issues will not become obvious until you've rolled up your sleeves started to use the product. And that's when many start to discover issues — *after* you deploy. There are several important items, but the single biggest challenge today is finding talent to help program and manage big data.

Talent, or Lack Thereof

One of the principal benefits of big data clusters is the ability to apply different programmatic interfaces, or select different query and data management paradigms. This is how we are able to do complex analytics, and how we derive better analyses options from the cluster. The problem is that you cannot use it if you cannot code it. Today's programmatic interfaces mean you need programmers, and possibly data architects, who understand how to mine the data.

Those of you who are building a security analytics cluster from scratch, should not even start the project without an architect to help with system design. Working from your project goals, the architect will help you with platform selection and basic system design. Building the system will take some doing as well as you need someone to help manage the cluster and programmers to build the application logic and data queries. And you will need someone versed in attacker behaviors to know what to look for and help the programmer stitch things together. There are only a finite number of qualified people out there today who can perform these roles. As we like to say in development, the quality of the code is directly linked to the quality of the developer. Bad developer, crappy code. Fortunately many big data scientists, architects, and programmers are well educated, but most of them are new to both big data and security. That brilliant intern out of Berkeley is going to make mistakes, so expect some bumps along the way.

Policy Development

Big data policy development is hard in the short term. As we mentioned above you cannot code your own policies without a programmer — and possibly a data architect and a statistician. There is a trend in the big data industry — not just security but for all use cases — to strap on abstraction interfaces to simplify big data query development. The goal is to minimize reliance on programmers, and get more non-technical stakeholder participation, but we are not there yet.

However there is good news if you are acquiring a solution from your SIEM vendors: most leverage SQL and the same policy management interfaces that you're already familiar with. You can continue to develop policies as you do today, without needing to retrain staff of bring on board new people.

But keep in mind this is a tradeoff: Using SQL interfaces keeps things simple at the loss of some query capabilities. Most SIEMs continue to use a SQL interface for their big data cluster, and hide the programatic interface with dashboards and visualization tools. The big advantage is you need not re-train your personnel to use the platform, nor find programmers and statisticians to get your project started. The downside is some of the powerful query tools and programatic methods

to search data are not at your disposal. You get a all of the performance and scalability benefits, and a majority of the search benefits, but not the level of customization and control you get from a standard NoSQL distribution. Whether this is a good or a bad thing will depend entirely upon your needs.

Policy Efficacy

One more facet of this difficulty merits a public discussion. If your SIEM vendor is introducing big data for the first time, you need to vet existing policies. With a radical shift in data management systems, it is foolish to assume that a new (big data or other) platform will use the same queries, or produce exactly the same results. Test queries on the new platforms with known data set to verify they yield correct information. As we transition to new data management frameworks and query interfaces, the way we access and locate data changes. That is important because, even if we stick to a SQL-like query language and run equivalent queries, we may not get exactly the same results. Whether better, worse, or the same, you need to assess the quality of the new results.

Data Sharing and Privacy

Many customers we spoke with want to leverage existing (non-security) information in their security analytics. Some are looking at creating partial copies of data stored in more traditional data mining systems, with the assumption that lower cost commodity storage make the iterative cost trivial. Others are looking to derive data from their existing clusters and import that information into Hadoop or their SIEM system. There is no 'right' way to approach this, and you need to decide based on what you want to accomplish, whether existing infrastructure provides benefits big data cannot, and any network bandwidth issues with moving information between these systems.

If you are considering moving sensitive data into your big data cluster, consider how you intend to protect it. This was not a question with traditional SIEM — both because the security model for relational databases was different and because big data is now leveraging more types of data. But your choice will be to secure the cluster itself, as we will discuss next, or to **apply data security controls**. Several firms we spoke with are using *tokenization* to substitute sensitive data prior to loading it into the cluster. This replaces original sensitive data with a proxy value that looks like the original, keeping data secure. And tokenization technologies provide a means of referencing original data values — de-tokenization requests — if needed. Some firms use *masking* to strip out sensitive data while retaining value for analytics; this choice is growing in popularity the sensitive data is often needed for data mining purposes. Others use *format preserving encryption* for specific columns of data, keeping it encrypted within the cluster, but enabling access to the sensitive data as needed.

Which option to choose is not easy to answer — it depends on how you want to use the data, and requires identifying a solution that will actually scale well enough to meet your needs. Still, if security of sensitive data is at issue, it is often easier to secure the data *within* the cluster than the cluster itself, given the current state of big data security.

Big Data Platform Security

NoSQL platforms generally offer poor security. The security features built into Hadoop are neither complete nor well thought out. With the exception of a couple commercial big data vendors who bundle security tools into their solutions, out of the box you do not get enough control to secure a NoSQL cluster.

If you're purchasing a big data solution from your SIEM vendor, you'll need to consult with them on how to secure the cluster so it meets your security and regulatory requirements. When deploying your own cluster for security analytics, you'll want to consider the following types of security controls:

- **Data Encryption**: To protect data at rest, ensure administrators or other applications cannot gain direct access to files, and prevent leaked information from exposure. We recommend file/OS level encryption because it scales as you add nodes and is transparent to NoSQL operations.
- **Authentication and Authorization**: Ensure that secure administrative passwords are in place and that application users must authenticate before gaining access to the cluster. Developer, user, and administrator roles should all be segregated. These capabilities are built into some distributions, and can link to internal directory management systems.
- **Node Authentication**: There is little protection from adding unwanted nodes and applications to a big data cluster, especially in cloud and virtual environments where it is trivial to copy a machine image and start a new instance. Tools like Kerberos help to ensure rogue nodes don't issue queries or receive copies of the data.
- **Key Management**: Data encryption is only as strong as key security; so use an external key management system to secure keys and, if possible, help validate key usage.
- **Logging**: Logging is built into Hadoop and many other clusters. It may seem nonsensical to log system event data when using big data as a SIEM, but consider the security of the cluster as distinct from the security of all other network devices and applications. We recommend that you enable built-in logging or leverage one of the many open-source or commercial logging tools to capture a subset of system events.
- **Network Protocol Security**: SSL or TLS is built-in or available on most NoSQL distributions. If privacy is at all important, look to implement protocol security to keep your data private.
- **Node Validation**: Leverage tools to pre-configure, patch, and validate nodes before they are added to the cluster to ensure baseline security. Most customers we spoke with use it in virtual or cloud environments, which offer incredibly simple tools for pre-deployment validation.

Closing Comments

There is a tremendous amount of hype around “big data” today, which makes security professionals worry that vendors are promising more than they can deliver — or even than big data is capable of. But the myriad different NoSQL environments have proven their worth at companies such as Google, LexisNexis, Netflix, and Amazon. Each of the major NoSQL distributions has demonstrated that it offers one or more major advantages in terms of scale, performance, data input, analytics, advanced query capabilities, data type management, or cost reduction compared to classical relational and flat file databases. There is no real question that big data works, but getting real value from its capabilities for better security analytics is not simple. Whether buying a solution from a SIEM vendor or building your own from scratch, success depends on careful platform selection and quality personnel to build the data mining capabilities.

If you have any questions or want to discuss your particular situation, 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 25 years of industry experience. He brings 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.

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 Nexus is an online environment to help you get your job done better and faster. It provides pragmatic research on security topics, telling 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 with 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. We maintain our strict objectivity and confidentiality. 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/>.