Interested in learning more about security?

SANS Institute InfoSec Reading Room

Security Management Systems: An Oversite Layer for Layers of Defense

We'll be discussing ways to make IDS and "traditional" security solutions more effective by "rolling up" security event information into an overall view of your organization's security stance. We'll also be discussing systems that take that a step further into providing either automatic prevention capabilities or that put security teams within a few clicks of implementing containment or preventative measures to minimize/mitigate the impact of (potential) attacks. The primary topic will be Security Event Management (SEM...

Security Management Systems: An “Oversite Layer” for Layers of Defense

Dan Keldsen

GSEC Security Essentials
Practical Assignment Version 1.4b
Option 1 – Research on Topics in Information Security

August 2003

Abstract:

The true strength of a well-planned and maintained information security solution is not just in “layered defense.” Layers are built primarily by breaking networks into subnets, De-Militarized Zones (DMZs) and so on, with multiple layers of screening routers, firewalls (or typically a single perimeter firewall), implementation of Virtual Private Networks (VPNs), anti-virus solutions at the desktop or server-level, and often, a healthy sprinkling of Intrusion Detection System (IDS) to (hopefully) identify (but not prevent) any malicious traffic not caught by perimeter defenses. These are narrowly focused solutions, with no awareness of the other layers.

More modern layered defenses consist of hybrid solutions that provide “layered” solutions in themselves (such as IDS combined with Vulnerability Assessment, or statistical analysis combined with pre-defined rules), forming more complete solutions with fewer gaps in the stacked security layers. Think beyond single task solutions – to “holistic” approaches, built to support the business rather than simply solve individual security problems.

We’ll be discussing ways to make IDS and “traditional” security solutions more effective by “rolling up” security event information into an overall view of your organization’s security stance. We’ll also be discussing systems that take that a step further into providing either automatic prevention capabilities or that put security teams within a few clicks of implementing containment or preventative measures to minimize/mitigate the impact of (potential) attacks.

The primary topic will be Security Event Management (SEM) and Security Information Management (SIM) solutions, otherwise known as Security Management Systems (SMS) – the next generation of centralized security logging, with the addition of powerful filtering, highly configurable notification options, knowledge of the layers of defense and protected assets, and in some cases, auto-prevention facilities, or automatic suggestions for policy change that human security operators would be implement.

Introduction – The Path We'll Follow

To get to the SMS discussion, some groundwork will be laid at first, to put us all on the same footing.

We'll begin with a quick discussion of the state of life on the Net, moving to a bit of depth on IDS and what it is (and more importantly, isn't) good for, where centralized logging has come from, some of the failures of components of "old style" centralized logging and the issues and resulting pain of building such a system yourself, and then for the majority of the paper, discussing the high-level as well as detailed points of SMS systems, and why these systems are important and **real** tools for improving your organization's overall security.

Recent History, and Fear of Change

The bad news is that the volume of attacks is increasing more rapidly than anyone would've reasonably expected – perhaps following Metcalfe's Law which roughly states that the possible cross-connections (and value of those connections) in a network grow as the square of the number of computers in the network increases). As the network grows, and the number of threats to the network components grows, the exponential growth of risk (likelihood of successful attack) in this environment is staggering – the benefits of Metcalfe's Law become a significant liability at times. [1] As to why attacks are growing larger in volume as well as sophistication, I will leave to someone else to discuss. It is a simple fact that the Net is both a great enabler for positive impacts on business and life, and a downright **dangerous** place.

The good news is that security tools are becoming more mature and capable at dealing with the world of threats that jump straight through firewalls (mimicking legitimate transactions) and the "traditional" tools of information security. "White hats" are catching up to adapt to the plethora of freely available tools for attacking in use in the "black hat" community – many of these "black hat" tools are extremely powerful and ingenious in circumventing security solutions.

So, how bad is it out there on the mean streets of the Net?

According to a recent EarthWeb article: "The world record for overt digital attacks in one month was broken Tuesday (20 May 2003) -- just four months after the last record was set and less than a month since the record was set for most attacks in a single day." [2]

So does anyone in the IT world honestly think it's getting to be a safer computing world out there? Of course not. While scare stories don't directly help make you more secure, it is certainly useful to make people aware, on a continual basis, that "secure" is not a static state. Even old, or at least well-known attacks and

vulnerabilities **still** remain unaddressed, due to lack of awareness, as well as lack of concern in allowing attacks to remain “in the wild.”

Even the most paranoid of us occasionally get lulled into a false sense of security – it’s tiring work to stay constantly on alert after all, but that’s why we need better tools to deal with the volume and severity of modern attacks, to provide the breathing room to get out of firefighting, constant reactive mode, and step back and plan solid security policies and systems as well as educating end-users, management and others.

You (the entire organization, not just you the lone security guy or team) need to be vigilant and always willing to look at your security systems with an eye towards what is **actually** working to make your organization more secure, and what is **not**.

Security investments aren’t made just for kicks – they should support the overall business or mission of the organization, and if the investments don’t help you make progress towards doing something other than chasing alerts, then something is seriously broken in your organization.

If it ain’t broke, but it ain’t workin’ either, what’s it for?

As mentioned earlier, an IDS by itself is not a protective layer. It **detects – period.** The resulting detection **may** provide information that allows future attack prevention, based on analysis of the resulting alerts. Then again, it may alarm so often that your security team can no longer keep up with it – and many of those alarms may be false.

So how are companies combining other security/management capabilities with IDS to make it more directly and immediately useful? Or in providing enough information to be within a click or two of reacting to an attack (auto policy suggestion), if your company doesn't have the stomach (yet) for automatic prevention?

First – to acknowledge the shortcomings of IDS before we go about “fixing” those issues.

Anyone who has used an IDS knows that they can be noisy – more realistically, they simply ARE noisy – the mathematical models that form the basis of these systems are prone to noise generation [3]. This does not mean that they are useless, or that IDS vendors have terrible programmers – but that you need to be aware of the limitations of these systems.

While it is quite straightforward to write new rules for Snort for example, it is incredibly easy to write rules badly – that inadvertently catch far more traffic than intended (false-positives). The magic of wildcards and regular expressions…

power can be dangerous, and there are subtleties in these systems that only experience can teach.

Second - Tuning IDS sensors appropriately to your systems probably takes more time than reacting to legitimate threats that have succeeded in intruding into your systems.

Unfortunately, constant manual tuning has no endpoint - you are never done, as no system lives in a static state of “secure” or (hopefully) “insecure” for long. Most IDS solutions are rules-based, and have no awareness of what assets (desktops, laptops, servers, networking equipment, firewalls, and so on) are in need of watching (why bother watching for Microsoft exploits if you are an all Linux shop) or what the current level of vulnerability is (if they are up to date with the latest patch from Microsoft or other vendor).

A slight tangent, as I’m a firm believe in humor to help get a point across:

(Taken from an advertisement for Computer Associates SMS solution [4])

Now marketing of course simplifies reality, or all marketing messages would be book length, but this point should be acknowledged. Large-scale IDS deployments (among other solutions) function much as the graphic above indicates.

How many fire alarms do you need, to let you know there is a problem? And what happens if your alarming system doesn’t differentiate between a short-circuit (false-positive) or “the real thing?” What if your alarming system **never stops ringing?**

If your alerting/logging solution isn’t providing you useful information to keep raising the level of security in your organization, then it’s time to take a good hard look at what you’re using for security, what you think the use of the tool is, and whether the tool actually directly applies to the task(s) at hand.

To continue – in my opinion, IDS, by itself, is for three things:

1. Auditing (does my security setup actually block what I believe it does?);
2. Education (Snort for example is an excellent learning tool), and;
3. Forensics (something got through, did I catch what it was and can I make sure I'm protected against future attacks of a similar nature?)

…and it is great for those areas.

But, if the primary goals of security are to enable business or prevent bad things from happening in the first place, then IDS has been very much misunderstood in the market – as people seem to expect that by liberally sprinkling IDS throughout their network that instantly, security is improved.

As the analysts at the Gartner Group say every few months: "Intrusion detection systems (IDS) are a market failure, and vendors are now hyping intrusion prevention systems (IPS), which have also stalled." [5]

I don't think either side of that statement is quite right, but I'd agree that in many cases IDS has been deployed for the wrong reasons, and too broadly. The herd mentality got everybody into this mess, and once in, it's hard to back out.

As far as IPS goes, yes, there is certainly hype and confusion on this topic (such as where it should be implemented – network, host, application, what techniques to use to identify attacks – behavior, statistics, anomaly detection, and what methods to block attacks – on the device affected, upstream, downstream, scrubbing the content to remove just bad portions, etc.), but we're just getting started in the "smart prevention" area.

IPS does however hold the promise of filling in both the necessary holes for "good traffic" in currently deployed layered defenses, as well as to address the issues of maintaining security in the face of completely untrusted or uncontrollable environments – such as wireless "hot spots," broadband access for guests at hotels, and so on. Stalled vs. failure seem like two differing scenarios to my eyes. Perhaps the lack of a budget for **any** new technology is a key to the stall? Hmm. But I digress…

It may seem that I have it in for IDS, but that's not the case. I'm against using the wrong tool for the job at hand, whatever the job or tool may be.

In any case, for those companies using IDS – a few ways to improve proactive rather than reactive uses, or at least speed up reaction time:

1. Tune the Rules - Manually, Constantly – Repeat (tuning IDS rules to minimize info-overload – the basic complaint about IDS). Most organizations don't have

the staffing to do this, and really, computers are supposed to do this sort of repetitive work so we can do "honest knowledge work," right?
2. Aggregate and correlate alerts with SEM/SIM solutions (do the heavy lifting programmatically – the primary topic of this paper… we're almost there)
3. Behavior (i.e. non-rule-based, or at least not JUST rule-based – to address "zero-day attacks" and variants that rules would not catch)
4. Auto-tuned to actual assets and vulnerabilities of those assets (no need for manual tuning, but high-level views of the organization aren't an outcome of this by itself)
5. Deception (i.e. honeypot/honeynet-type features to pick off the easily identified attacks/threats and therefore minimize intrusions and resulting alerts. Also used to waste the time of an attacker on attacking a false/deceptive target, while buying time for the defender to gather information to thwart the attacker once they turn to the "real" target)

I had begun writing this paper thinking that I would tackle all of these areas, and I believe all of them require some serious research and discussion to hash out what is good and bad about each approach.

There is no single path to "being secure" – so I'm simply attempting to raise some further issues for thought that others can carry forward.

Nonetheless, the rest of this paper will focus on centralized logging, and specifically, the benefits of using a Security Management System (SMS) rather than just any old centralized logging solution.

Centralized Security Logging: Where We've Come From

To get the broadest picture of the state of security in your organization, logging needs to be centralized.

That may or may not literally mean centralized to one location at the head office – that would be determined by your organization's security policy, and specifically whether there is a need/want to centralize security logging on a regional basis (NorthEast Region – Boston, New York, and Providence offices for example), or within a single office (all network/security logs in the Boston office sent to a central logging system in that office).

You may perhaps want a multi-tiered solution, that allows the above scenarios, but also forwards security data to the head office for the 50,000 foot view of the Chief Information Security Officer (CISO) and his/her team.

In any case, centralized security logging is not a new idea. I'm not going to cover a lot of old ground here, but will instead refer to two papers in the SANS Reading Room: one on centralized logging for Windows Internet Information Servers (IIS), and another which discusses the pros/cons of an entirely syslog-based logging system.

As for the IIS server paper, the writer's goal was to consolidate the Event viewer logs, Internet Information Services (IIS) logs, and Urlscan logs from 15 Windows 2000 web servers into a database he could query against.

The results of the queries would then be automatically emailed to him once a day for easy perusal when he arrived at work. While his solution wasn't specifically looking at overall network security, and was instead focused on watching IIS servers via three independent logging sources, he ran into a number of issues that took quite a bit of ingenuity to work around and that reflect some of the basic issues involved in doing centralized logging well.

Some of the issues raised: [6]

- For the three different source types, three different applications had to be used to send/export the data from the source to the central server, each with their own syntax and features.
- Batch files needed to be created for each of these applications, and stacked in such a way as to not overlap with each other or collide with nightly backups – and due to the size of the files, this was only done once a day, after hours.
- Entirely different log formats meant reformatting and creating different SQL queries/scripts to import the information into the centralized SQL Server.

- Unique SQL statements, hard-coded for specific error codes and statements, were built for the reporting mechanism.
- Due to the volume of the raw data, data was only kept in the SQL Server for a day, making historical reporting or future trending nearly impossible, unless summarized reports were kept and used to "diff" (compare) over time. Useful in a pinch, but not very flexible.
- While the solution provided a way to check security on a daily basis, it was focused strictly on forensic, or "post-attack" analysis – although he states that he was also in conjunction using a few other tools to do real-time alerting from the three different event sources as well. This solution was meant to double-check the real-time alerts to see if anything was missed.

To expand this solution to handle other sources and log types, all of these steps would need to be repeated again, resulting in extra work in manipulating the raw data before yielding useful end results. It works, but it takes a lot of analysis and hard work to get it running, and from the author's example, easily a dozen tools simply to move the events to one location and do fairly limited reporting.

As far as Syslog-specific concerns go, the summary of security issues for "normal syslog" implementations are: [7]

- Transmission of system log data is in the clear (sniffable)
- Use of UDP for network transfer (no verification that data was received)
- Storage of event data is in clear text (modifiable and searchable)
- The sender and receiver are not authenticated (spoofable)

In short, syslog is a quick and dirty method for sending logs around, and is supported by nearly every modern operating system. Unfortunately, just because it "sends events easily" does not mean that it's an appropriate foundation for your security logging system.

The referenced paper [7] goes on to suggest next-generation versions of syslog that incorporate using TCP rather than UDP, encrypted network transfer (SSH, Stunnel or IPSEC), and using database-backed storage such as MySQL and encrypting and protecting the database(s) with appropriate standard database tools.

These solve some of the issues of syslog, and we'll run into these concepts later as well, in the discussion of SMS solutions (as syslog may feed into the first tier of your SMS solution), but issues of securing the transport and storage of security events are only one concern of many that SMS solutions address.

So, even with a thoroughly modern "secure" syslog-based logging infrastructure, and a centralized place to put logs, we have tackled only one (or two) pieces of the puzzle.

Once you have the data in one place, the real power comes in manipulating that data so that it is **useful**, rather than just a collection of stuff that nobody will ever look at, and most importantly, acting on that data in real-time (or darn close to real-time at any rate – due to some of the features that you will want from an SMS there will be some minimal amount of delay – see discussion of aggregation and correlation later in the paper for further info).

A quick quote from an InfoWorld article on ArcSight (a SIM vendor):

> In the good old days, it was enough to write some scripts that would parse a handful of log files and have a human translate those results into an assessment of the situation. But the growing dependency on always-available, interconnected systems; the large number of ways these systems can be compromised; and the shortage of people skilled enough to interpret evidence of these compromises makes it increasingly difficult to get a grip on just what's going on both inside and at the edge of the corporate network. [8]

To summarize these issues:

- Large security solutions produce raw information overload
- Real-time analysis is needed before damages of attacks overwhelm us
- Systems have complex relationships, that may be nearly impossible to fully document completely – making troubleshooting difficult
- Lack of skilled personnel necessitates "smarter tools"
- Identification of real threats versus noise is nearly impossible as a result of all of the above

The unsaid aspect of the problem here is that the parsing of data is not only a complicated business (recall the issue of writing rules for IDS in the first place), but that the syslog-based centralization approach is nearly always running scripts on a periodic basis – more to the point, **not in real-time**.

As indicated in the centralized IIS logging paper previously [6], the purpose is to have a reporting mechanism that runs once a day to identify how the network and the machines within it were compromised, rather than attempt to stop the attack in the first place. The very nature of this setup is nearly the opposite of what is needed to maintain a steady state of security. To prevent insecurities from destroying your environment, a real-time, transaction-based, "intelligent" (machine-assisted) security management system is key. Centralizing the events is certainly important, but as an actual security measure, it is both too little and too late.

Rules – Enough to make you cry…

I started off this paper discussing some of the issues of IDS, specifically false-positives and the dangers of poorly written rules. If you were to trod down the path of the home-grown centralized syslog system (or the central data store of your choosing), you would be further subjecting yourself to a second layer of rule-writing at the central server to further cut down the raw logs from the IDS sensors to a reasonable volume for human consumption.

Not only do you need to keep the sensor rules up-to-date (a tiring task by itself), but the rules at the central server that do further filtering also need to be updated. It's inevitable that the code base between the two layers of sender (the IDS sensor) and central collector is going to get out of synchronization – and if this task gets sufficiently out of hand, many man-hours will be wasted identifying where the code has fallen apart and consequently NOT be spent on making security happen.

Remember, the task at hand is to make your organization more secure, not to delve into the intricacies of Regular Expressions and scripting languages – unless those skills serve you well in other capacities.

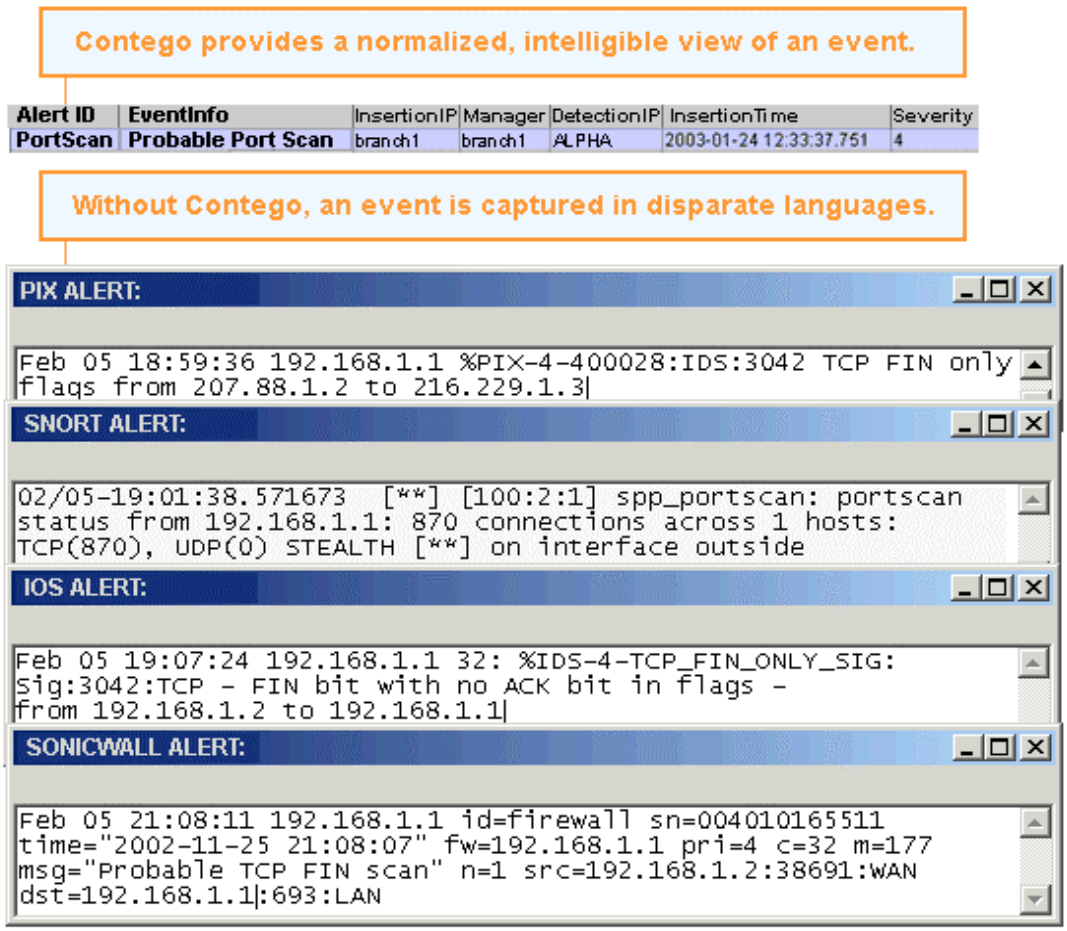
A further piece of the centralized logging puzzle – there are no (real) standards for the data that is sent by the devices/systems you are logging. Standards are being pursued, but are far from universal adoption.

Specifically, there are no standards for:

- Order for the data (the order or number of fields)
- Names of the fields (to assist in puzzling out the order)
- Data within the fields (is a HIGH alert the same as a LEVEL 5?)
- Naming conventions of known attacks or potential attacks (although there a number of publicly available databases of attacks such as MITRE's CVE that are frequently referenced by security systems)

That means that any rules that you write to identify a stealthed port scan (designed to avoid tripping firewall rules and identify ports for later exploitation) for example, may have to deal with a Cisco PIX alert of "TCP FIN only flags from x.x.x.x to y.y.y.y" while Snort says "status from y.y.y.y: x connections across 1 hosts: TCP(870), UDP90) STEALTH [**]" and a Cisco IOS-based device says "TCP – FIN bit with no ACK bit in flags – from x.x.x.x to y.y.y.y." All of which are stating the same basic concept "a stealthed port scan detected" – but which not only have different names for the attack, but have the information in different orders, and with different delimiters in the text.

To illustrate this – with events shown in normalized form versus individual systems alerts:



(TriGeo normalization demonstration documentation [9])

Writing effective rules and scripts to make sense of this information can by itself be a full-time job, and while it is certainly POSSIBLE to do it yourself, perhaps there are other, more directly USEFUL tasks you could be doing for the organization, that again, contribute to the state of security in the organization more directly and immediately.

Security Management Systems (SMS)

So, now that we’ve discussed the pros and cons of typical layered defenses and centralized logging, we are at the primary topic of this paper, and an area that is rapidly moving from an interesting idea to a near requirement in maintaining security systems…

Let’s tackle some semantic issues, so we’re speaking the same language. Here’s what I’ve been able to uncover about the differences between the SEM and SIM definitions:

1. Security Event Management (SEM) is more in keeping with "traditional" IDS, in the sense that it is more forensically focused (i.e. the attack has happened, but at least you know about it; you don’t have as many alerts to deal with, and you should have the data to show how it propagated through your network). Events are collected in one place, with duplicates removed and normalization of the data to make the alerts understandable by non-specialists.
2. Security Information Management (SIM) is said to be focused towards proactive security by way of coordinating logs quickly enough to enable human intervention, with potential intrusion prevention built-in as a capability or enabled via connections with (for example) OPSEC (Check Point's API to enable 3rd parties to interact with Firewall-1) and other APIs to Firewalls, routers, host-based prevention measures, etc.. It’s built on top of the basic SEM functionality, and adds further intelligence – and an opportunity to for the system itself to prevent attacks.
3. Depending on the vendor, flip that around.

Some vendors specifically state that SEM is just the capturing of all these events into “one pane of glass” and it’s still up to humans to figure out what they means, while SIM correlates across devices, **and** adds some intelligence – the context as to what these events actually mean, specifically an attack, questionable, or normal behavior.

In any case, the terms are a bit meaningless, as they are so interchangeably mentioned that you really need to look at the details of what the system offers and decide whether it does what you want, regardless of what they’re calling.

The primary functionality is the same however, they are built to roll-up logs streaming for various security solutions - primarily IDS and firewalls. Inclusion of audit/system logs from Windows systems (NT and beyond), as well as routers, and other devices that output SNMP, syslog, and similar streams are possibilities as well.

For the remainder of the paper, we’ll simply call these systems Security Management Systems or SMS, unless a specific product name identifies itself as SEM or SIM.

Now, while there are some open-source tools that address the individual features that are an integral part of full-blown SMS – you’re in for a serious bit of hacking the tools together to reap the full benefits of a pre-integrated SMS. Recall the steps described and number of tools involved the fellow needed just to get event into one place with simple daily reporting [6]. It’s not a quick hack to pull off.

For large networks, or networks with more than a handful of device/system types, the need to scale to handle hundreds to thousands of devices, and high-bandwidth environments, this means that commercial grade tools are essentially a requirement.

I would be happy to be proven wrong on this, as I am certain this type of solution is something any business or large organization is going to **need** simply to survive, and pricing these solutions at the high-end leaves far too many holes in the fabric of the Net to my liking.

Raising the low-level of security across the board simply makes it that much more difficult for the “black hats” of the world to find easy targets and wield them as zombies or otherwise inflict damage on the rest of the world.

So, why choose to add SMS to your toolbox?

Primary complaints of IDS are the problem of false-positives (reporting a problem an attack when you are not vulnerable or simply misinterpreting normal traffic as an attack) and false-negatives (not reporting an attack due to outdated signatures, badly tuned signatures, inability to keep up with volume of network traffic, etc.).

SMS solutions are in a **unique** position to qualify whether traffic really is “bad” or "normal” that other solutions are not – context based on multiple data points, from across multiple layers, which enable if not 100% certainty, at least sufficient evidence to warrant further analysis.

This technique is what differentiates an SMS’s decision capabilities from, for example, a host-based intrusion detection solution (HIDS) that uses behavior-based modeling to flag “abnormal” behavior **at that point in the environment**. As mentioned earlier, making decisions based on single pieces of information, single analysis techniques or isolated placement of sensors within the environment can cause false-positives, false-negatives, or simply blindness to attacks.

Realistically, if you have more than a handful of security solutions deployed, you need something to group the logs to make sure you aren't missing an attack simply due to being overwhelmed by disparate and disconnected logs from the various devices and systems you have. With the current rate of attacks, you

cannot possibly react quickly enough when looking at multiple sources of data in an after-the-fact manner.

Moreover, SMS is useful for normal day-to-day operations, incident handling and post-attack forensics - to capture summary information and (depending on the SMS solution) detailed packet captures or even full playback of traffic (assuming you have the storage space available to keep that level of information at hand).

By pulling together security streams in a holistic manner (example: watching packet pass the outside IDS, router, firewall, DMZ, inside DMS, Windows-based web server), there is more information available to make the good/bad determination than from a single device point of view.

Rather than simply looking at whether a firewall rule has allowed or disallowed the traffic (for example, web servers obviously need to be reachable, so ports 80 and 443 are typically open on a firewall), by catching the logs fired off by various security-specific and general network or system devices, an SMS has the opportunity to at the very least paint a more complete forensic picture to determine what happened, and also to put in place further (and more effective) layered defenses.

Now you may be saying, “But wait, I already have Internet Security System’s (ISS) SiteProtector or Enterasys Networks Dragon Enterprise Management Server to do this for my sensors. Why do I need another SMS product?”

For the most part, SMS solutions that are delivered by IDS or other single point security solution vendors are not built to handle the breadth/variety of devices that most networks have deployed. They (unsurprisingly) focus on support of strictly their own devices, or at most a handful of third-party devices.

For IDS vendors moving into this space, it is also likely that they will take a while to move from the basic SMS functionality (log roll-up and removal/grouping of redundant alerts) to also applying preventative security measures (such as changing firewall rules, adding Access Control Lists (ACLs) to routers, and so on).

Until IDS vendors themselves have an Intrusion Prevention System (IPS) offering, they would be pointing out deficiencies in their own arsenal by tying themselves into OTHER security solutions with Intrusion Prevention-like abilities, which seems an unwise move to secure greater market share or simply survive in this economy at all.

Perhaps, if you are an entirely Cisco shop, it may be possible that you could use their solution (CiscoWorks Security Information Management Solution [10] - which is actually an original equipment manufacturer (OEM), licensed version of netForensics v3.1 SIM solution [11]) to roll-up your security events.

In this case, you **would** be getting a package that can handle quite a variety of devices, and that is also tooled specifically to handle Cisco's SAFE Blueprint - a security framework that Cisco espouses, using Cisco products as well as partners (including netForensics) for a holistic security framework.

This is not a bad option, technologically speaking. Cisco solutions are not known for being cheap, however. Purchasing and rolling up the support options directly through Cisco may have it's own appeal if you feel that there are benefits to gained by single-sourcing your security options – such as a volume discount or single company to call for support.

The Cisco scenario is the only exception to the rule that I am aware of at this point, and so my recommendations on SMS would be to look to third-party providers who are not also providers of IDS (in particular) or other security solutions.

For example, while I think Internet Security Systems (ISS) has some great security tools, and they have a SMS solution of their own, they are clearly focused on providing the range of security solutions themselves – so support for third-parties within their own SMS offering, seems awfully unlikely. On the third-party support page for ISS SiteProtector (their SMS), it only mentions Cisco PIX and Check Point specifically. [12]

Unless they have demonstrated commitment to and delivery of adaptors, agents, connectors, etc. for a broad array of the devices that you are concerned about, be wary. With the current state of the economy, it is far more likely that companies will play it safe by focusing inward on their own products rather than branching out to others.

Another aspect to be aware of for SMS solutions is the issue of "scalability." I have not been able to find definitive tests of exactly how these solutions stack up against each other in handling: events per second, raw throughput, accuracy in collapsing the event stream as they purport to do, etc.. I would certainly like to take their word for it "they are highly scalable," but that seems like quite a leap of faith.

The only third-party research I was able to find that had hard numbers for performance was an eWeek review of Network Intelligence's Engine LS Series [13]. "In tests, when we used a log event simulator to put the LS Series (a DSRV, an A-SRV and a Local collector) through its paces, the cluster easily handled the 10,000 events per second we threw at it." They don't provide much detail on exactly how the testing was set up, and a simulation introduces many variables that differ from "real systems," including network topology, existence of other "real/normal" traffic and the impact of attacks on that traffic, is the alert stream **really** generating the volume they purport, etc..

Unfortunately, lab tests aren't realistically going to be able to recreate the environment that YOU have, and so it behooves you to do your own trials, in the exact environment it will run in. This would include simulating attacks as well as watching for any currently existing attacks, and ensuring that the SMS solution can effectively handle your current needs as well as to reserve capacity for systems under attack (such as a Nimda or SQL Slammer attack may provide).

If you are a suitably large customer (or really, anyone considering these systems), demand a "bake-off" amongst your choice of vendors, to see exactly how these solutions work on the **exact** same traffic, at the same time. At the very least, work out a 30-day trial of the technology so you can get through installation and initial setup hiccups, as well as have enough time to get some real hands-on experience with the solutions to make a thoroughly educated decision when you are ready to make the final purchasing decision.

Deployment of SMS

These solutions can be installed on COTS (commercial off the shelf) hardware (i.e. a standard Windows, Unix, Linux, or other OS machine), or on dedicated appliances. If deployed on COTS hardware, you have two options, depending on the vendor you choose:

- Install the basic OS and environment yourself before installing the SMS on top of that
- Installing a pre-hardened/configured OS (such as Linux or a stripped-down Windows 2000 installation) with the SMS from a CD-ROM

This choice basically comes down to whether you trust yourself of the vendor to adequately secure the overall system, and also whether the vendor you've chosen gives you any choice in the matter.

The desire to minimize initial costs may justify re-purposing existing hardware, or buying relatively low-end PCs. However, appliance-based solutions may offer greater reliability, lower TCO (Total Cost of Ownership), smaller installed space requirements (1-2 space rack units), and the benefits of pre-built and hardened systems - helping to ensure (although there are no guarantees) that your security solution doesn't become a security problem in itself.

Personally, I find appliance-based solutions to be my first choice, as I have other things to do than to configure a secure Windows or Linux installation and then further install a security solution on top of that. I would much rather spend my time getting the solution tuned to the task at hand than spend time in pre-configuration issues. Your mileage may vary.

Once You're Logging, What are You Watching For?

Keep in mind that using a SMS solution can also reveal insider problems, such as attempts to modify or access accounting or human resource systems, not just to detect and stop "Evil Hackers from Russia."

Specifically, by feeding logs from file servers, databases, desktop/laptop operating systems (Microsoft Windows-based typically, although some SMS solutions address UNIX OS environments), and internal firewalls or screening routers, your SMS can uncover malicious activity that isn't crossing your outside perimeter – or that has bypassed perimeter solutions via desktop modems or a successful Trojan installation.

Remember the previous discussions about the problems of layered defense? This is why a view across layers is necessary! **ALL** layers, not just the outward-facing perimeter layers.

While reports are conflicting about whether insider activity or outside "hackers" are the primary source of security problems, it still behooves you to maintain vigilance on both sides of the external firewall. An attack is an attack – regardless of which direction it comes from.

Even though mention constantly arises about "the soft center in your Tootsie-Pop security" – most security attention is still focused at the "traditional" perimeter offerings – firewalls, routers and VPNs, specifically. IDSs are used as a last-chance alarm for inbound attacks, and **might** also see outbound attacks depending on sensor placement, but otherwise, there is typically a severe lack of data on (possible) malicious activity on the inside of organizations.

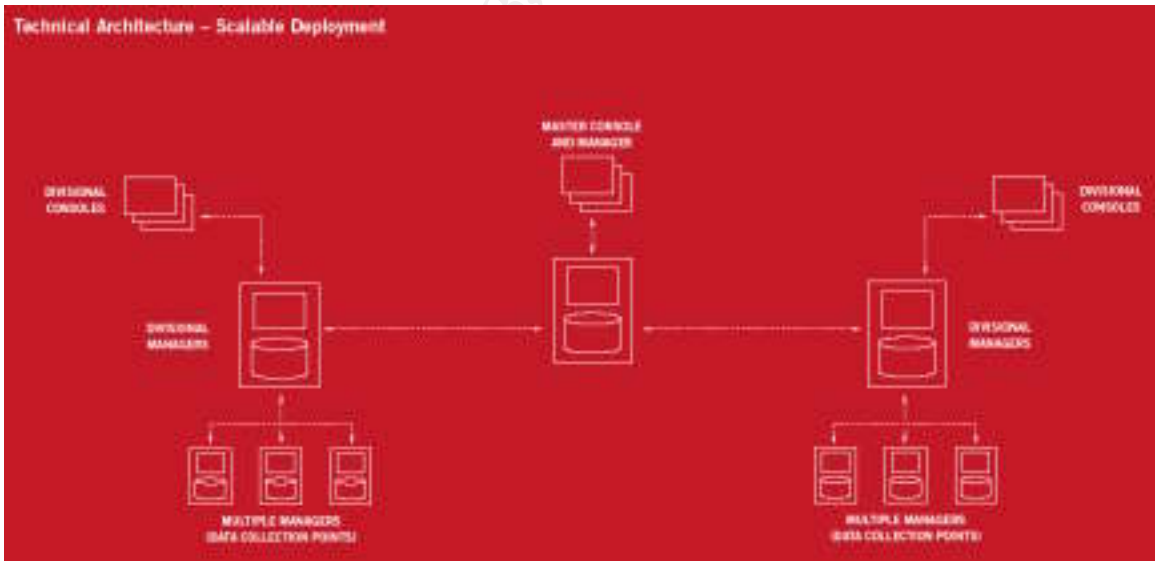
Some of this is due to the nature of most organizations to trust those people you know (employees, contractors), and the desire to not violate that trust by imposing oppressive logging and blocking/preventative policies. In most cases however, it is simply that the interior network changes far more drastically and quickly than the relatively limited array of applications/services running through the perimeter defenses, and specifically that the interior network does not stay static long enough for a policy to be adequately defined in the first place.

By using an SMS to roll up security data from **all** corners of your organization, you should find that you can raise your security stance significantly across the entire organization, rather than continuing to harden the outside and leave your underbelly exposed – trusting that your users are well behaved or not inadvertently introducing viruses, worms, and so on.

What are the Components and Features of SMS?

Architecture

- Three-tier – the standard architecture
  - Collection agent – either installed on the device or receiving data sent directly to it via syslog, SNMP, XML, OPSEC, POP and other methods
  - Manager – receives data from agents and manipulates the data to do correlation, act on correlated information
  - Console – real-time and historic reporting and configuration management application (or browser-based)
- Multi-tier – Three-tier architecture expanded to distribute load or handle Division to Enterprise management roll-ups (see graphic below)
  - Collection Agent
  - Divisional Manager
    - Divisional Console – for local viewing of SMS
  - Master Manager
  - Master Console
- Two-tier
  - Collection agent and Manager in one system/appliance – data is sent directly from the relevant devices to the central server
  - Console – same as three-tier



(Diagram from ArcSight Documentation [14])

In some SMS solutions, the database/storage repository for event collection is built-in to the manager component, and in other cases, utilize a deployment of standard SQL servers such as Oracle, Sybase, Microsoft or open-source variants such as MySQL – constituting another functional tier. Pricing may or may not

include the cost of the database environment itself. This is one area where your own research into the appropriate solution to your situation needs to reflect these potentially hidden costs. Buyer beware!

For the most part, all SMS vendors claim that a DataBase Administrator (DBA) is unnecessary since all of the DB manipulation is handled by the SMS itself, but as a review from the labs of InfoWorld magazine states, that is not always the case:

> Getting ArcSight up and running is not a trivial task. Although the basic OS and memory requirements are fairly straightforward, your Oracle database must meet specific requirements. We'd like to see future versions support IBM's DB2 and/or Microsoft's SQL Server, if only because in our own case, Oracle's canned startup scripts proved defective. [8]

In any case, a database is necessary for an SMS to do its magic – and you may not have a choice of database vendors – some SMS solutions only support one vendor (typically Oracle). If that is an issue for you due to internal policy, politics, existing vendor relationships, and so on, then your choices are going to be a bit narrow. On the flip-side, it really shouldn’t matter – as long as it works and doesn’t cause additional pain, this choice is probably the least of your worries.

Secured Communications

All SMS providers that I’m aware of secure the communications between their own components – sidestepping one of the issues discussed in the homegrown approach. In many cases, the security includes SSL-encrypted communications, as well as digital certificates for all of the components, to prevent sniffing and injection of false information into the system.

The eagle-eyed among you may recall that I had mentioned in the beginning that “traditional syslog” approaches leave many things to be desired. However, many SMS solutions still use syslog, SNMP, and other unsecured and unauthenticated channels to get their data in the first place. Conundrum?

As I also mentioned (and referenced in the recent paper on syslog issues [7], there are alternatives available to create “secure” syslog solutions, and sidestep this particular hurdle. There are also many other methods of doing data collection, including modern solutions built with secure transport in mind – such as Check Point’s OPSEC APIs, and Cisco’s POP protocol. Just keep these things in mind as we proceed and when you’re discussing concerns with the SMS vendors.

Data Presentation:
Real-time, Historical Reporting, Forensics, Replay

Any SMS should be focused on real-time – that's the main point of these systems, to remove delays in reacting to attacks or suspected attacks.

Historical reporting is useful for trending information as well as the expected weekly/monthly/quarterly/yearly reporting for senior management as well as for auditing and reporting purposes, and again, is a standard feature for all SMS systems I've run across. Beneficial extra features for historical reporting purposes are scheduled reports, that can be run on nearly any schedule you desire, output in a variety of formats (PDF, HTML, delimited text variants, as well as Excel), and deliver output to local drive, network drive or e-mail to a recipient list.

Forensic analysis can be considered the same as historical reporting, although it would be targeted at gathering evidence on specific events rather than overall reporting of the environment. Some SMS solutions (such as ArcSight) have built-in case management functionality, which allows a forensics analyst to both find evidence and write-up the incident (or case) directly within the environment. Clearly, not everyone needs this, or can afford it, but minimizing the numbers of tools for those who do, is a great way to speed up these tasks.

From a usability standpoint, having the ability replay traffic can be very useful in both a "near real-time" standpoint (replaying recent traffic to track down an issue) as well as for post-event forensic purposes.

All of the SMS solutions I've come across provide both real-time and historical reporting facilities. However, playback, particularly visually represented (rather than in streams of tables), is a relatively rare feature – ArcSight and netForensics being the highlights.

ArcSight provides the same interface regardless of whether you're looking at **current** (now) real-time data, or playing back previously captured data – which again lets the security team focus on security rather than in learning multiple interfaces to the same data. Data flows into a personalizable dashboard with both tabular and graphic displays of security information, including a display the call the "event radar" that consolidates the current levels of attack categories in a display resembling a digital graphic equalizer – useful for a quick glance at the current status of your systems in very little screen real-estate.

NetForensics takes visual playback a step further via integration with SilentRunner's 2D and 3D visualization tool. Currently data has to be exported from NetForensics and pulled into SilentRunner, but they have live integration between the two systems on the near-term roadmap.

Note: SilentRunner is worth a look to see the likely direction of visualization for SMS systems and network monitoring in general. It may not be affordable for everyone, but for the extra visualization capabilities it provides, there are certainly some organizations that will see that it can shortcut response times significantly in complex investigations. [15]

Aggregation

Note: For many of the following terms, there are again some discrepancies in how the vendors label and described these functions. I found that descriptions offered [16] by OpenService (www.open.com), a SIM vendor, were the clearest, and so I echo and expand on their definitions in many of the following sections.

Aggregation is also called "event compression" by some SMS vendors. This is an initial round of event processing, typically at the first tier – the collection agent – although some systems save this step until events are delivered to the second tier – the manager – to do both aggregation and correlation (see next section) in one place.

Simply put, aggregation is used to cut down the number of alerts sent across the network for a few reasons:

- To process the data as close to the source as possible to lessen the load that the central manager has to process
- To do initial reduction of multiple similar alerts (500 portscans in 2 seconds) to a single summarized alert, and depending on vendor, add some of the initial intelligence or act preventively on this information
- As a side benefit, to minimize traffic on the network (security solutions shouldn't be a burden to your infrastructure), and avoid sending redundant copies of log data in raw form (which can be very large streams indeed)
- This also makes the overhead of encrypting this traffic (to prevent revealing the SMS to attackers) that much lower, since there is less to encrypt at any given time

Aggregation depends on a time-based threshold, which depending on your needs, you may need the ability to set multiple thresholds for various types of attacks, as well as the ability to keep an eye out on the "low and slow" attacks that are frequently mentioned as theoretical possibilities but not seen as much as outright attacks.

A time-threshold is necessary to determine when to start and stop counting multiple events and send along a summarized event to the next level of processing.

Aggregation can be limited to identifying and compressing alerts either on a single device (firewall x reports 500 port scans from x.x.x.x), or in some cases

from a single “class” of device, such as all firewalls (firewall x, y, and z report 2000 port scans from x.x.x.x). Ultimately, all SMS solutions do aggregation, and whether they do multiple levels of aggregation may be unimportant, assuming that multiple-device and specifically multiple-classes-of-device correlation happens further down the processing chain.

In some cases, it may not be desirable to remove duplicate information through aggregation – such as instances when complete forensic evidence is needed to pursue legal actions, during periods when completely unknown “zero-day” attacks are appearing, or for occasional “deep audits” where greater details may be wanted to thoroughly audit policies, procedures and systems and verify the effectiveness of these areas.

For the greatest flexibility, some SMS solutions offer the ability to selectively turn on/off aggregation, based on alert types, currently alert levels, time of day, and so on. Not everyone will need or want this flexibility however.

Normalization

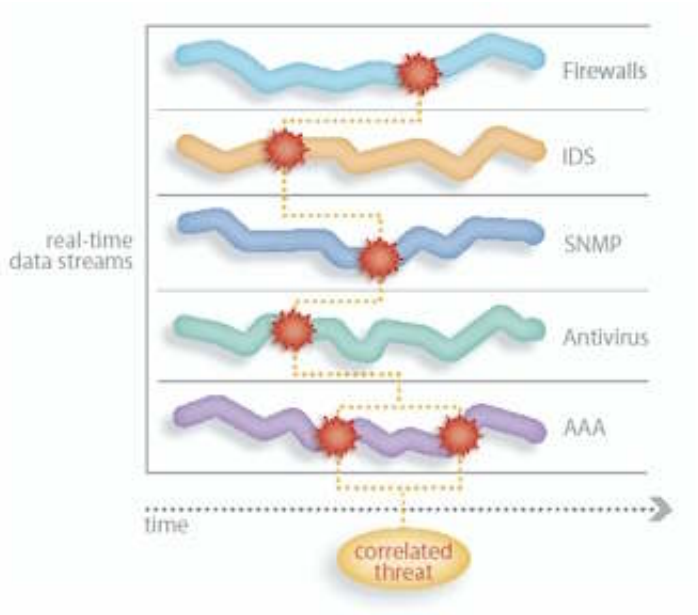
As mentioned previously in the overview of SMS, normalization essentially standardizes the presentation of alert/event data into a single format, for both data manipulation purposes (the intelligence of the SMS), as well as for human readability. Without normalization, SMS is useless, and this is again an area which is NOT addressed by homegrown or single-platform/device oriented centralized logging systems.

Amongst vendors, there are many claims about the level of normalization which is required and handled by their solutions. Some normalize to 6-12 fields, stating that their research shows that is where 80-90% of the value is provided, and that the extra overhead of storing more isn’t worth the minimal extra benefit. Others take the opposite approach and normalize data to nearly 100 fields or more – assuming that if there is data buried in there that will help at all, it is best to keep it as a potential data-point, rather than toss it.

Obviously, the more data at hand, the more you have to work with. However, the greater your storage requirements, the more processing that is necessary to find and manipulate the data, and so on. Much of this discussion depends on your exact needs and the budget for both the SMS that provides this and the resulting storage and CPU requirements and sufficient backup/redundancy capabilities. This can get quite expensive very rapidly – so be careful to work through these scenarios to fully identify the costs with the vendors or integrators/resellers you are talking to. With large of volumes of data at hand, you are also likely going to want full-fledged DBAs to handle the intricacies of managing large datasets.

Correlation

So, just to further clarify exactly what correlation means in a multi-device network is a graphic to illustrate the point:



(Diagram from OpenService [17])

So, correlation is essentially taking aggregation to the next logical step – comparing events received across multiple devices/security solutions, and the resulting alerts (or events of interest), and determining if 2 or more matches are found to be related to each other, and thus constitute a correlated threat. This is again a time-based and time-sensitive process, and for maximum flexibility you should be able to set your own time thresholds at least for the overall system, if not for individual classes/categories of attacks.

Automated correlation is one of the primary benefits of a SMS – and is one of the most difficult things to do as a human, particularly if the interface to discover these events in the first place is:

- Buried in individual logs/consoles for each security solution – forcing the user to switch between various windows to find
- Simply dumped directly into one master location – again up to the user to spot the important data
- The result of numerous pages or screen alerts coming from different security solutions, with no clear indication of the relation between the events

Threat Calculation

Once a correlated alert has been found identified, there may still be too much data coming in to narrow the field for your security team as to which alerts are the most dangerous. To add extra value, an SMS needs to be able to prioritize the alerts based on awareness of your environment.

For example, while it may indeed be the case that a Code Red attack has gone through the firewall (it's web traffic after all), and past an internal IDS sensor, and thus the SMS has correlated and classified this as a real alert – if you don't run IIS-based web servers, or they are already patched against Code Red, then why should you care about this alert?

Most people shouldn't care (unless you have a need to monitor every conceivable intrusion), so in the best case scenario, an SMS should be aware of the assets in the environment (i.e. IIS server, Apache Server, Cisco routers), or the vulnerabilities in the current environment or ideally, both – to make sure there isn't a gap between the understanding of the environment implied by the two data sources. An extra component in determining the severity of a threat is what value is held by an asset – what is the value of an asset being online and functioning properly versus under attack and down? You may have "high value" or "mission-critical" servers that absolutely cannot go down, in which case any attack on those system should be the highest priorities. There are however, many nuisance attacks that don't carry significant catastrophic damage, and the threat level should reflect that even against the "high value" systems, unless it's going to cause severe damage, that the threat level should stay at a lower level, for follow-up as time allows.

Some SMS systems provide multiple (near infinite in some cases) methods to classify the assets and their importance in your organization, and some are limited to one or two. Your needs should dictate your choices here as well – and much of this discussion may be the subject of internal politics or "turf wars" between network operations and security operations, as well as server or desktop support areas and management from each contingent.

From a pure capability standpoint, it makes sense that being able to specify a dollar value, and classify assets by the location, operating system, running applications, business owner, business unit, and so on are very useful features that help to further narrow down the information that your team will be acting on and prioritize responses. This is the same sort of exercise involved in building a taxonomy for a content management system, or in laying the groundwork for a data-warehousing/data-mining system – the more finely-grained you can slice the data, the more ways it can be analyzed and combined, but if having this capability makes **your** situation more complex, then it may also be wise to strive for the simplest solution.

Intelligence/Knowledge/Enrichment

Once the SMS has gotten to the point of nailing down a bona-fide alert and ranked the threat appropriately, the further facility you will find useful is the ability of the SMS to identify exactly what the issue is by tying into resources provided by any of several methods:

- Vendor(s) of the device(s) under attack (for example links to Microsoft's Knowledgebase for a IIS-related attack)
- Links to third-party databases such as the previously mentioned MITRE CVE database/website
- Built-in knowledgebases provided by the SMS vendor
- Knowledgebase built up by your own personnel within the SMS or other system

Security solutions are notorious for spouting acronym-laden or abbreviated and numerically coded alerts, which is fine for database search purposes, but unless you have a very sharp memory or ability to decode these alerts on the fly and what they mean to you, is not much help to the humans using these systems.

Turning the alerts into human-readable information and with further human-readable information about what the implications and solutions to the issue are, is where SMS solutions start to add the highest value of all.

Notification/Action

Once an alert has gotten to this stage, something should happen with that information. Notification/action options can be:

- Pop-up alerts on an operators desktop
- Console-alerts (inside the console/application that the operator is presumably logged into)
- Pager notifications
- E-mail notifications
- Creation of a ticket in a system such as BMC's Remedy (www.remedy.com) – for follow-up by help desk personnel outside of the SMS mechanisms
- Addition of source, destination, or type of attack to a "watch list" for further narrowed alerting/reporting
- Creation of a "case" to begin the process of forensic investigation, or to add to an existing case of a similar nature
- And execution of scripts or applications that do whatever your heart desires – perhaps firing off an SMS message, AOL Instant Message, send a fax, trigger a lockdown, etc.

Notifications and actions can get quite complex, or be as simple as only having one choice. All SMS solutions have some notification facilities, or it there would hardly be any point in having identified a problem.

At first glance, it would seem that perhaps a simple e-mail or console notification would be enough options. An alert comes in, a notification goes out, repeat at the next alert, meanwhile a person acts on the notification, and problem is solved. In the real world however, you may have periods where alerts are coming in fast and furious, and this relatively simply notification method would start burying the operators in alerts via whatever mechanism was chosen.

Event suppression

Through aggregation and correlation, "alert storms" should already be minimized, but in the case of long, sustained attacks, the time-based thresholds that aggregation and correlation depend on may be exceeded over and over again, generating fresh alerts each time.

To combat that problem, and again reduce the issue of having "noisy" security systems, event suppression adds functionality that essentially silences or slows alerting while similar alerts are being generated, to prevent burning out the user and related notification systems by essentially creating a Denial of Service attack on itself. Until you experience it yourself, this may not sound serious. When an attack is underway, and your pager is buzzing every second, while e-mail piles up in your inbox, and the SMS console is flying by with too much information, you'll have an entirely different opinion.

Not every SMS solution has this feature.

Acknowledgement

To close the loop on notification systems, you need to at **least** be able to acknowledge receipt and possibly action on the notification, to make it easy to glance at the system console and see what alerts have not been dealt with. Remember, we're trying to simplify our lives here, and that means letting the system help us by freeing us from maintaining a memory or scrap of paper with the current alert status.

To my knowledge, all SMS offerings have this feature.

Shift-rotations and Awareness

Further, awareness of shift rotations (night vs. day, weekday vs. weekend/holidays, $1^{st}/2^{nd}/3^{rd}$ shifts) is useful, if needed for your situation. This allows the system to only notify the relevant parties, and leave others alone. This

is particularly important to avoid burning out your best personnel by always having one person or team be the first responders to every single alert. Paging the most expensive, qualified or over-worked team member at 3am in the morning for a minor alert while someone is sitting right in the data center is perhaps not the smartest method of notification.

Not every SMS solution has this feature.

Escalation

Escalation is a further extension of notification. If a notification isn’t acknowledged within a pre-determined timeframe (again, for maximum flexibility, variably set by you to specific attack types), escalation could attempt notification again to the same person via the same method, try another method for the same person, or move through a list of alternative people to contact through the various methods specified. Again, this may be impacted by awareness of shift-rotations so that the Level 1, 2, and 3 support within a specific shift are notified first before moving to alternatives.

Not every SMS solution has this feature.

Notification – The Bottom Line

Not everyone needs the maximum flexibility here, and it may be that you already have a notification and ticketing solution in place, and the SMS should simply hand-off the information to that solution. These issues will certainly impact the choices you have for an SMS, or lower the cost of your solution if you are able to price out the removal of sub-components you already have or don’t need.

Is the end in sight?

At this point, the flood of alerts has been narrowed by reducing the volume (aggregation), turning the smaller stream into useful data for the system (normalization), tracking the threats across multiple devices/systems (correlation), classifying appropriately to prioritize response (threat calculation), identifying exactly what it means and how to deal with the problem (knowledge/intelligence) and letting someone or something what has transpired and how to solve it (notification).

If you are not interested in automatic intrusion prevention or containment, at this point the SMS is done and humans take over and solve the issue, hopefully to never see the same alert at a later point.

Prevention/Automatic-Action

Mmost SMS providers have **some** facility for firing off not only alerts to humans at this point, but to activate a script, influence a firewall or other security device, or otherwise directly enable some sort of preventative measure. This is where people begin to get nervous – and most SMS vendors do not dwell on this aspect of their abilities since it is somewhat controversial.

Intrusion Prevention Systems (IPSs) or systems that act like IPS, make people sweat bullets – the idea of preventing an intrusion simply fills most people with dread.

What if it's not really an attack? What if it's the CEOs daughter, trying to IM him to find out if her new Mercedes is in yet? Or what if it's a brokerage firm's best customer, placing a $5 million (USD) market buy order seconds before the close of market? Or the main website suddenly receives 10,000 hits a second, due to being linked from the homepage of CNN.com – and looking for all intents and purposes like a Denial of Service (DoS) attack?

You aren't paranoid if they really are out to get you…

Well, in some cases, this is a leap of faith to believe perhaps, and in other cases, it's a legitimate concern – to fear the ramifications of auto-prevention gone haywire.

In many cases, it's a case of people being burnt by relying on the information streaming out of poorly tuned and rampantly deployed IDS – and the mess of bad information (false-positives) that they've experience as a result.

As I hope to have illustrated by this point however, an SMS is feeding upon quite a bit of data to draw it's conclusions, and while it is still possible that it is labeling normal (or at least not bad) traffic as an attack, it's far less likely to do so than a single solution/sensor is.

Prevention Taken Seriously

I have only run into one vendor who is really taking the IPS-like potential of SMS to heart and pushing it out to the world.

TriGeo's Contego SMS solution features what they call "Active Response" and is built with the premise of their outlook on security that they term "NATO5" – taken from the NATO alliance, and the 5th article of the charter that states "an attack on one is an attack on all." [18]

So, in this case, they're moving from beyond "awareness of an attack is a call for eventual human action" to "an attack is happening, let's stop it immediately."

I performed an analysis of TriGeo's solution and specifically covered their "Active Response" mechanism in late July/early August 2003 [19]:

> TriGeo's vision is to provide enterprise protection "from the Perimeter to the Desktop" – which is a tall order. Thanks to their ability to tie into a variety of security solutions such as firewalls, anti-virus solutions, and IDS, as well as their own Secure Point of Presence (SPOP) software agents, the Contego solution offers a comprehensive solution that actually covers the ground that they claim.
>
> The SPOP model that TriGeo uses provides not only a locally-installed engine to forward logs to the Contego Manager (a standard SMS feature) and for viewing by a sysadmin at a Contego Console (again standard for SMS), but also provides an enforcement point for additional proactive security.
>
> This allows Contego to enable functionality that would normally be reserved for a host-based IPS (HIPS) solution, but with the added benefit of awareness of the larger picture provided by the SIM-side of the solution.
>
> Key to the SPOP functionality is in applying encrypted, two-way communications between SPOPs and the manager – this ensures that logging and active response are protected from abuse/attack and don't constitute an additional security concern for your organization.

While the secured channel between agent and manager is typical and expected of an SMS, it is **absolutely necessary** when the channel is used to implement automatic prevention. The potential for an intrusion prevention system to go haywire is a significant reason that (potential) buyers of such solutions are wary of turning them on. If attackers could manipulate this facility themselves, then there would be no further need for them to find other vulnerabilities on your systems – they could simply turn your own tools against you.

> An example of SPOP abilities: Contego notices an attempt to quarantine a virus has failed. Contego can direct the SPOP on the infected desktop to disable network connectivity until the system can be cleaned, and alerts the admins, preventing widespread infection without requiring heavy integration into network switches, routers or firewalls or needing manual action to physically unplug the system.

This is exactly my point about the benefits of SMS versus individual solutions. Many anti-virus tools only report errors to the user sitting **at that machine.** It would then be up to the user to let the relevant team/personnel that the virus

engine had failed, and then action would have to be taken. The delay between detection or alarm and action is at the least minutes, and potentially, infinite – as users are far more likely to simply ignore the error and go about whatever they were doing.

By raising the alarm and sending to the SMS, action can be taken, and in this case, **automatically** as defined by pre-defined policy within the SMS, extending the capabilities of the security solutions you already have in place and narrowing the gaps in your defenses. With new viruses and Trojans attacking much more quickly and aggressively then ever seen before, action needs to happen as quickly as possible to prevent massive infestations inside and outside the organization.

Supported Devices/Systems

I began this paper discussing the issues of IDS and homegrown centralized logging solutions. A primary concern of IDS, home-grown, and commercial SMS, is that these systems need to be able to handle the full gamut of devices or systems that you have deployed, or at least the subset that you care about including in your overarching view of the enterprise.

If an SMS vendor does not support the essential items from your environment, that SMS is not useful to you, as the point is to provide a comprehensive view of your security. Punching holes in your coverage is almost worse than having no view at all or going back to managing multiple, independent consoles.

However, web-sites and PDF lists of supported devices from SMS vendors frequently run a bit behind in giving you the latest list of supported equipment, so ask the vendor directly on the phone, via e-mail or in person, for the very latest list, and if need be, apply pressure to get support for the equipment in question.

Most SMS vendors have an SDK or API, perhaps even a Wizard-Driven agent creator that you can use to create your own collection agents. Your mileage may vary, as I haven't run into any research indicating successful deployment of these customer-customized agents, but that doesn't mean it hasn't been done.

The ability to support a wide variety of equipment is a competitive differentiator, so in many ways, it's in the best interest of the SMS vendors to be able to grab data from as many areas as possible.

Twist an arm, plead your case, create a petition – the economy is in tough shape at the moment, and if these vendors aren't going to get a sale just because they are unwilling to dedicate a few man-minutes/hours/days to work that is reusable for other customers, you should take your business elsewhere in any case. The right answer is either "yes, it's in there" or "It will be in the next release."

Delving into Highlights of Selected Specific SMS Solutions

Creating a full-fledged buyers guide with exhaustive details on all SMS solutions is not what I’m aiming for in this section. Instead, you will find a few SMS vendors listed, and highlights of what are unique, or in some cases, missing features, offered across the range of solutions.

Some interesting tidbits have been garnered from discussions with customers of these companies, webinars I’ve attended and their Q & A discussions, conversations with the vendors, or via research done on the web and from magazine articles and reviews.

Pricing details can be hard to find from companies offering SMS solutions – and are highly variable depending on size and scope of your deployment and any need for professional services. The economy being what it is at the moment, prices are, shall we say, flexible – to a lesser or greater extent depending on the company you’re dealing with. Where I’ve been able to uncover pricing, I’ve included it.

ArcSight

Pricing: As mentioned in an InfoWorld Test Center Review: "Pricing depends on the number of consoles and monitors deployed, so the typical setup could cost anywhere from $75,000 to $250,000 and even higher." [8]

Confirmed pricing with ArcSight – although the low-end **could** be as low as $50,000.

ArcSight's architecture consists of 3-tiers (as well as distributed model mentioned in the earlier **Architecture** section):

1. SmartAgent
2. ArcSight Manager
3. ArcSight Console

(see http://www.arcsight.com/graphics/product/archdta.pdf for more detail)

Normalization: They normalize to over 70 fields and greater than 100 categories of attack. Claim to capture 100% of the original alarm or alert from the device or sensor – which is why their normalization schema is so large. This is far more data than any competitors, and while perhaps overkill in most cases, is an interesting philosophical difference from the other vendors – no data is unimportant.

Aggregation: Can be selectively disabled – by time of day, type of alert, and many other ways – again to maximize the full capture of **all** data. This obviously increases bandwidth requirements as well as processing time at the manager, but again, if needed, it's convenient to have this flexibility.

Deployment: Their solution is Java-based on all three tiers, which should allow the components to run on any OS capable of running the Java Virtual Machine (JVM). Officially, support is for Windows, Solaris, RedHat Linux, and AIX.

List of supported devices/vendors:
http://www.arcsight.com/graphics/product/SupportedProductsDatasheet.pdf

Universal Agent: Custom SmartAgents can be built by the end-user if your devices are not officially supported. Caveats mentioned previously apply here.

Visualization: Live dashboards, 2D and 3D graphic representations – including their unique "Radar" display, bar charts, pie charts, and others. Modes: Real-time, Replay (ability to replay captured traffic through the same interface rather than just as a static report), Replay with Rules (same as replay but with added wrinkle of applying SmartRules to model solutions to historical threats without impacting current traffic).

Database support: Currently only supports Oracle for the event repository.

Case/incident management: **(Note – very powerful. Appears to be the most complete of all SMS vendors.)** Can be done either completely within ArcSight, with an external trouble ticket system (such as Remedy), or both. Cases are dossiers containing event information, context-sensitive knowledge base pages (an html content management system that can house organization policy and procedures, device vendor information, CERT advisories, etc.) and security analyst comments on the case.

A complete reporting structure is set up to track case metrics (how long have them been open, what the stage of resolution is, cases per analyst, etc.) is also included. Cases can automatically be opened or added to as the result of a correlation rule firing, or information can be packaged and send it to a system like Remedy.

Scoring/Threat Correlation: As mentioned in the earlier **Threat Calculation** section, ArcSight uses multiple criteria to calculate and rank threats. "ArcSight TruThreat Risk Correlation" factors in: severity of the threat by itself, asset value (in dollars) of system under attack, vulnerability (from reports imported from third-party vulnerability scanners) of the asset to this attack, and correlated alerts across the logs being fed into the system to help minimize false-positives.

Reporting: Standard reporting features available, outside of the real-time interface, for historical reporting purposes and forensic investigation. Additionally, ArcSight provides what they term "Delta Reports." These reports essentially allow an organization to do periodic "baselining" (establishing normal traffic patterns/statistics) and use a few of the statistics from this baselined snapshot to compare against "live" traffic. "SmartRules" then compare the two (baseline vs. live), and trigger alerts (user-defined) based on statistically significant deviations from the baseline. This raises ArcSight away from the some of the shortcomings of a strictly rules-based approach and introduces the beginning of anomaly detection. I would imagine other SMS vendors would take this approach (eventually) as well, since a strictly rules-based approach to alerting at the SMS level suffers from the same issues as at the individual sensor level.

Correlation actions: Console notification, send page/e-mail, execute a command (any executable or script – potentially preventative measure), change severity of rule (to raise alarms more quickly), create new case or add to existing case, add source of attack to suspicious list or "hostile zone" (virtual grouping).

ArcSight and CERT – Global Outreach

An interesting offering of ArcSight, recently (August 2003) announced in conjunction with the CERT Coordination Center (CERT CC), is similar in (basic) functionality to the Dshield.org project that most SANS followers are already aware of.

Called the Cyber Security Information Sharing Project (CSISP), the underlying concept is to use data collected from three (not yet named) universities, using the new distributed architecture ArcSight has built in conjunction with CERT CC.

Attack data is funneled directly from the participating universities ArcSight implementation, scrubbed to remove identifying or sensitive data prior to sharing it “up” into the CERT CC at Carnegie Mellon in Pittsburgh, Pennsylvania. [20]

In some ways, this collaboration has seems a bit limited, as it is initially going to be a installed strictly at CERT CC and only three universities, whereas the Dshield.org project is open to all that care to participate.

However, this is the first deployment of this system, and no doubt, significant traffic will be seen by these participants. In any case, this is a nice bit of outreach that is fairly rare these days, although ArcSight obviously stands to gain publicity and ultimately drive sales as a result of this arrangement.

The sanitization and packaging of data from this offering also points to future possibilities in enabling companies to do automated reporting of intrusions that expose “private information” (that could facilitate identify theft) to customers and authorities, as required by the California Law “SB 1386” [21]. For mandatory reporting needs, having tools that support that need could be a major productivity gain as well as a way to stay in compliance with the law and avoid being fined by governmental agencies.

I found another interesting commentary on this move, from a security blog [22]:

1. While ArcSight is standards based, it's not clear from the announcement that the initiative will offer a means by which participating organizations can use software from vendors other than ArcSight which adheres to the IETF standards -- IDMEF (Intrusion Detection Message Exchange Format) and IODEF (Incident Object Description and Exchange Format)
2. There is still the issue of a centralized consolidation and analysis of data. While there is no doubt that the initiative can assist in the (near) real-time submission and analysis of attack data from distributed locations, it still makes the community highly dependent on CERT and only CERT.

So, echoing my own thoughts, this author is somewhat dubious of the motive, but hopeful that this is a good sign. With any luck, this move heralds the continuing opening up and sharing of information on the “white hat” side of the fence for both users of this data, and providers of tools to do this sort of collection and dissemination.

Interestingly enough, this is one of the first visible steps in the Homeland Security initiatives that are meant to open up information sharing amongst commercial, government and other organizations.

While SANS and many other organizations have been leading the charge on raising security awareness and skills to new levels, there is quite a bit of room for improvement, as the “black hat” community continues to outpace us in the dissemination of new attack methods.

By taking information sharing up at the commercial level (via ArcSight’s involvement) and non-profit level with the Carnegie Mellon CERT CC, it certainly feels as though we may be on the verge of getting a much larger group of people participating as “global watchdogs,” which will make it that much easier for all of us to stay ahead of the future attack waves. Hopefully.

netForensics v3.1 (also available as Ciscoworks Security Information Management Solution)

Pricing mentioned from the June 11, 2003 netForensics webinar [23]:
Starter pack - $40k (30 devices, Oracle database license, Engine)
20 more devices - $20k
Additional engine - $20k
Additional database - $10k (for backup primarily or as an alternative location for simultaneous analysis – at a regional deployment as well as enterprise-wide for example)

Marketing Mantra:
Normalize, Aggregate – the events from across the enterprise
Correlate, Visualize – Identify & respond to threats in real-time

netForensics' architecture consists of 3-tiers (as well as distributed model mentioned in the earlier **Architecture** section):

1. netForensics agents – normalizes data at the agent before feeding up to;
2. netForensics engines – which correlates data, adds information, passes up to;
3. netForensics SIM Desktop – where alerts are shown, reports are run, and where management of agents and engines occur, etc.

(see http://www.netForensics.com/documents/pr_architecture.asp for more detail)

Deployment: Their solution runs on RedHat Linux, Solaris, Windows variants on COTS hardware, or as mentioned later, on a Cisco/netForensics appliance.

Collection: Agent-less - their deployment is typically watching streams of syslog, SNMP, etc. - installed agent mostly only necessary for Microsoft server installs.

List of supported devices/vendors:
http://www.netForensics.com/documents/pr_devices.asp

Universal agent: SDK to support other devices not directly supported by their collection agents. Caveats mentioned previously apply here.

netForensics' 9 categories of events:
1. Denial of services
2. Reconnaissance attempts
3. System status/configurations
4. Unknown/suspicious
5. Viruses/trojans
6. Policy violations

7. Application exploits
8. Authentication / Access / Authorization
9. Evasion

20,000 security event types from agents are normalized to roughly 100 ALARM ID's (in 9 categories above)

Visualization:
Out of box, targeted to security analyst, geographic view, enterprise view, management, but configurable for individual users needs.

250 pre-built report types - security, risk management, utilization, system

Reports can be drilled down into, by device, time, query, alert, etc.

Cisco/netForensics have been working together for 3 years now. NetForensics is an OEM partner as well as developer partner in maintaining support of Cisco devices.

Signature updates: They e-mail customers about new updates, who can then use central facility ("the provider" in console) that pulls down new signatures. No auto-install option available – customers didn’t trust and don’t want automatic action in this regard.

Incident management: Can integrate with Remedy - for escalation process and ticketing within existing help desk applications.

Database support: Oracle DB included in the starter pack. That's the only solution they found with appropriate enterprise reliability and scalability.

Benefit of Cisco relationship: Cisco IDS v4 with XML streams output - netForensics agent can already handle that, much easier integration and modern approach than syslog or SNMP.

Scoring: Doesn't rank threats against vulnerable assets, although they are working on pulling in vulnerability reports from other vendors, on short-term roadmap.

Report generation: can be pushed out as CSV, HTML, and PDF. Includes access to security portal - where reports can be posted rather than dynamically generated (for performance-reasons as well as historical reporting).

Their SEM/SIM definitions:

- SEM - deals with events only, aggregates events, displaying them on console, up to operators to sort through the mess

- SIM - adds intelligence, aggregate AND correlate, point out threats for you and how to deal with them

SilentRunner integration - just implemented first phase - outputting info to file that SilentRunner can import and playback. Next phase - SilentRunner directly accessing the SIMS database. SilentRunner provides extensive visualization capability, particularly topographical network view, also unique playback abilities- in 2D or 3D.

"Cisco/netForensics Offer Appliance"

Near the final publishing of this paper (July 31st, 2003), Cisco and netForensics announced the culmination and immediate availability of a jointly developed appliance-based version of this offering, called the CiscoWorks SIMS Engine 3.1. The Appliance is meant for regional deployment (in a single office for example), reporting up into the software-based CiscoWorks SIMS 3.1 system in the central office or perhaps in a managed security service provider (MSSP) control/war room. Pricing is $40,000 (list) for a single appliance. [24]

As far as my research has taken me, with this new addition, Cisco/netForensics, Network Intelligence and TriGeo are the only three options that have an appliance-based offering. Keep that in mind if the simplicity and pre-configured benefits of an appliance are a necessity for your operations, as your choices are awfully narrow with those constraints.

It's interesting that Cisco and netForensics chose to roll out a more narrowly focused (targeted at small/medium office) appliance for the first appliance – I would expect a larger "management appliance" to be in the works as well, although perhaps they feel that the management server is likely to be too customized to the individual customer needs to work as a viable appliance offering. Stay tuned for information on the next stages of this development relationship.

OpenService’s Security Threat Manager (STM)

Some aspects of this section are from an analysis I performed on STM in May 2003 [25].

Pricing: Not publicly divulged, although reviews put the typical enterprise installation price range at between $100,000 to $400,000. [26]

Threat Calculation: Weights threats according to their category/class and the target being attacked. Unique "decay" feature ensures paused or low-and-slow attacks are still raised out of the background noise.

Escalation: Pre-defined as well as custom escalation schemes available.

Knowledge: Workflow support to third-party sources (vendors, MITRE, etc.) links threat data to knowledge about the attack being used.

Console: Web-based console allows attack drill-down and bulk solution configuration in simple, single operations from anywhere. Pure HTML (no Java).

Device Support: Support for a wide variety of security technologies, from open source solutions such as Snort to vendors such as Checkpoint, ISS, Cisco, and Nokia

(see http://www.open.com/products/threatmanager/systemwatch.shtml for more detail)

Database: Uses embedded Sybase database – no further licenses needed.

Network management: STM is built on top of an engine that has been used for many years to do network event management, and therefore uses awareness of device up/down status as well as performance data to feed it’s correlation model. Other SMS offerings can be customized to take some of this information in, but STM is the only solution that I know of with this awareness built-in and integral to the product.

Health monitoring of STM: Unique “heartbeat” feature within STM architecture constantly checks the health status (up/down) of it’s own architecture – to make certain that the SMS itself is functioning as expected.

TriGeo’s Contego

Some of the information below was taken from an analysis I had performed of TriGeo in August 2003 [19], and also mentioned previously in this paper in the **Prevention/Auto-Action** section.

Pricing: Starting around $18,000 and scaling up dependent only on the number of devices being monitored. Unlimited consoles are included in the solution.

Deployment: Appliance-based solution for central management functionality allows rapid implementation and hands-off installation. Appliance (the manager) also includes Snort, for built-in IDS capabilities.

Consoles: Unlimited license for Java-based admin consoles allows flexibility in deployment to suit organizational structure and remote admin/monitoring needs

Device support: For an extensive list of commonly deployed technologies, from McAfee, Symantec, and Trend anti-virus, to Cisco, and Snort IDS, Check Point, Cisco, NetScreen, SonicWall and WatchGuard firewalls, and networked clients and servers running Windows, Linux, Netware, Solaris and Unix.

(see http://www.TriGeo.com/downloads.php?id=3 for more details)

Database: Built-in to appliance/manager, no need for separate license or platform.

Intrusion Prevention/Auto-Action: **Note: TriGeo takes the most aggressive and proactive stance on doing auto protection, so keep that in mind if your SMS solution requires IPS-like abilities.** Proactive security enabled by both “Active Response” (see earlier section on **Prevention/Auto-Action**) policies that influence dynamic firewall policies at the perimeter as well as Secure Point of Presence (SPOP) policy enforcement features on desktops and servers that can isolate internal threats to the environment.

Conclusion

The basic premise of good security is layered defense – the aim of which is to build complimentary (and perhaps redundant) solutions that prevent attacks from being successful, and thus "make your organization secure."

The SMS approach provides an essential "Oversight Layer," which correlates activity across the entire enterprise, and thus provides a more realistic real-time view of attacks to enable containment of damages or outright prevention.

Therefore, I believe an SMS is a necessity for all but the simplest of networks. SMS provides information to allow your security team to take **action** quickly, by providing links to information on how to fix problems (vulnerability remediation), suggesting explicit policy changes to your layered defenses to block identified threats (if the SMS supports that), by literally reaching out and stopping attacks automatically through built-in prevention capabilities (such as TriGeo's Contego) or via controls enabled between the SMS and other security layers.

One good indicator that a technology is gaining steam is coverage of the sector by Gartner Group, which occurred in their April 2003 "Magic Quadrant" on SIM: "The IT Security Management Magic Quadrant Lacks Leaders" [27].

While it may be somewhat early to call a "leader" of the SMS pack, the **need** for an SMS solution is real, and likely a necessity for any organization as we move forward with wireless access points, armies of users on laptops, broadband everywhere and so on. The currently offered consoles for each of those systems individually just do not provide the scope of information needed to handle enterprise-wide security as a whole – due to their limited views into the organization's security, as well as typical deployments that feed the consoles of these individual layers to entirely separate personnel who have no insight into threat activity elsewhere in the organization.

An effective SMS must enable you and your organization to get immediate, specific, human-understandable information that lets you prevent or quickly contain an attack, rather than force the security team to puzzle out where the links between events are, what they mean both individually and collectively, and what the resulting action should be.

It's time to consider Security Management Systems for your organization – to multiply the abilities of your existing security infrastructure, as well as adding an "Oversight Layer" that is desperately needed.

References:

[1] Keldsen, David. “Application of Metcalfe’s Law to the Identification and Prevention of Security Attacks.” Personal communication. 9 August 2003.

[2] Gaudin, Sharon. “May Breaks Record for Digital Attacks” 21 May 2003. http://itmanagement.earthweb.com/secu/article.php/2210321 (June 7 2003)

[3] Axelsson, Stefan. “The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection.” In Proceedings of the 6th ACM Conference on Computer and Communications 7, Nov. 2-4, 1999.

[4] Advertisement for Computer Associates eTrust Security Command Center: http://www.ca.com/etrust/management/ (4 June 2003)

[5] "Intrusion Detection Systems Earn Failing Grades From Gartner" 11 June 2003. http://www.techweb.com/wire/story/TWB20030611S0006 (12 June 2003)

[6] Lalla, Gregory “Centralizing Event Logs on Windows 2000” 28 February 2003. http://www.sans.org/rr/paper.php?id=902 (5 August 2003)

[7] Nawyn, Kenneth. “A Security Analysis of System Event Logging with Syslog” 27 June 2003. http://www.sans.org/rr/paper.php?id=1101 (3 August 2003)

[8] Connolly, Pat. “Zeroing in on threats” 28 February 2003. http://www.infoworld.com/article/03/02/28/09searcsight_1.html?s=tc (18 July 2003)

[9] “Consolidate Logs with TriGeo” http://www.TriGeo.com/consolidate.shtml (9 August 2003)

[10] CiscoWorks Security Information Management Solution (SIMS) http://www.cisco.com/en/US/products/sw/cscowork/ps5209/ (7 June 2003)

[11] netForensics v3.1 Security Information Mangement (SIM) Solution http://www.netForensics.com/documents/pr_sim.asp (7 June 2003)

[12] “SiteProtector™ Third Party Module” http://www.iss.net/products_services/enterprise_protection/rssite_protector/tpm.php (11 June 2003)

[13] Chu, Francis. “LS Series Logs Better Security.” 12 May 2003. http://www.eweek.com/article2/0,3959,1081315,00.asp (8 June 2003)

[14] “ArcSight Technology Architecture” http://www.arcsight.com/graphics/product/archdta.pdf (8 August 2003)

[15] Keldsen, Dan. "The Global Grid is Alive! (and infested…)" 5 December 2002. http://www.aiimne.org/dan_keldsen_delphigroup.pdf (9 August 2003)

[16] OpenService "Understanding Aggregation" http://www.open.com/solutions/Aggregation.shtml (8 August 2003)

[17] OpenService "The True Meaning of Correlation" http://www.open.com/solutions/TrueCorrelation.shtml (4 August 2003)

[18] "TriGeo Network Security – NATO-5 Technology" http://www.TriGeo.com/nato5.shtml (5 August 2003)

[19] Keldsen, Dan. "Security Information Management (SIM) - It's Not Hype, and Not Just for "Large Enterprises" 7 August 2003. http://www.delphiweb.com/knowledgebase/guest.htm?action=search&__woP_SECTIONSdatarq=39&__eqP_IDdatarq=2373 (8 August 2003)

[20] Fisher, Dennis. "CERT to Ease Sharing" 28 July 2003. http://www.eweek.com/article2/0,3959,1204398,00.asp (3 August 2003)

[21] Poulsen, Kevin. "California disclosure law has national reach" 6 January 2003. http://www.securityfocus.com/news/1984 (8 August 2003)

[22] Miller, Rich (28 July 2003) "CERT to announce data sharing initiative" http://www.telematica.com/blog/categories/security/2003/07/28.html (8 August 2003)

[23] McNabb, Dustin. Morrison, Randy. "CiscoWorks SIMS 3.1 Solution Briefing and Technical Overview " 11 June 2003 – 1pm ET. http://www.netForensics.com/documents/re_webcasts.asp?web=true&webinar=new (11 June 2003)

[24] "CiscoWorks Security Information Management Solution Datasheet" 30 June 2003. http://www.cisco.com/en/US/products/sw/cscowork/ps5209/products_data_sheet09186a008017dcb6.html (1 August 2003)

[25] Keldsen, Dan. "OpenService - How to Survive Drowning in Alerts" 16 May 2003. http://www.delphiweb.com/knowledgebase/guest.htm?action=search&__woP_SECTIONSdatarq=39&__eqP_IDdatarq=2341 (25 May 2003)

[26] Desmond, Paul. "OpenService Bundles Security Tools" 12 December 2002. http://www.esecurityplanet.com/prodser/print.php/1556211 (9 August 2003)

[27] Messmer, Ellen. “Gartner and SIM” 7 April 2003. http://www.nwfusion.com/cgi-bin/forum/gforum.cgi?post=312 (5 May 2003)

Backup Reference Materials (unused in this paper, but useful reading):

Fratto, Mike. “2003 Survivor’s Guide to Security” 15 December 2002. http://www.nwc.com/1326/1326f23.html (3 August 2003)

Fisher, Dennis. “NetForensics Gets a Face-Lift” 18 October 2002. http://www.eweek.com/article2/0,3959,640455,00.asp (8 June 2003)

Network Intelligence “Devices supported by Network Intelligence’s SMS solution.” 29 May 2003. http://www.network-intelligence.com/supporteddevices/ (8 June 2003)

Koziol, Jack. “Real-time alerting with Snort, part 1 of 3.” 12 June 2003. http://newsforge.com/newsforge/03/06/09/1939256.shtml?tid=2 (15 June 2003)

Last Updated: September 25th, 2017

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS DFIR Prague Summit & Training 2017	Prague, CZ	Oct 02, 2017 - Oct 08, 2017	Live Event
SANS Phoenix-Mesa 2017	Mesa, AZUS	Oct 09, 2017 - Oct 14, 2017	Live Event
SANS October Singapore 2017	Singapore, SG	Oct 09, 2017 - Oct 28, 2017	Live Event
Secure DevOps Summit & Training	Denver, COUS	Oct 10, 2017 - Oct 17, 2017	Live Event
SANS Tysons Corner Fall 2017	McLean, VAUS	Oct 14, 2017 - Oct 21, 2017	Live Event
SANS Tokyo Autumn 2017	Tokyo, JP	Oct 16, 2017 - Oct 28, 2017	Live Event
SANS Brussels Autumn 2017	Brussels, BE	Oct 16, 2017 - Oct 21, 2017	Live Event
SANS Berlin 2017	Berlin, DE	Oct 23, 2017 - Oct 28, 2017	Live Event
SANS Seattle 2017	Seattle, WAUS	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS San Diego 2017	San Diego, CAUS	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Gulf Region 2017	Dubai, AE	Nov 04, 2017 - Nov 16, 2017	Live Event
SANS Miami 2017	Miami, FLUS	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Milan November 2017	Milan, IT	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Amsterdam 2017	Amsterdam, NL	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Sydney 2017	Sydney, AU	Nov 13, 2017 - Nov 25, 2017	Live Event
Pen Test Hackfest Summit & Training 2017	Bethesda, MDUS	Nov 13, 2017 - Nov 20, 2017	Live Event
SANS Paris November 2017	Paris, FR	Nov 13, 2017 - Nov 18, 2017	Live Event
SANS San Francisco Winter 2017	San Francisco, CAUS	Nov 27, 2017 - Dec 02, 2017	Live Event
SANS London November 2017	London, GB	Nov 27, 2017 - Dec 02, 2017	Live Event
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZUS	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Khobar 2017	Khobar, SA	Dec 02, 2017 - Dec 07, 2017	Live Event
SANS Munich December 2017	Munich, DE	Dec 04, 2017 - Dec 09, 2017	Live Event
European Security Awareness Summit & Training 2017	London, GB	Dec 04, 2017 - Dec 07, 2017	Live Event
SANS Austin Winter 2017	Austin, TXUS	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Bangalore 2017	Bangalore, IN	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Frankfurt 2017	Frankfurt, DE	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DCUS	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Oslo Autumn 2017	OnlineNO	Oct 02, 2017 - Oct 07, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced