Vulnerability Management Evolution: From Tactical Scanner to Strategic Platform

Version 1.3
Released: May 15, 2012

Securosis, L.L.C. 515 E. Carefree Highway Suite #766 Phoenix, AZ 85085 T 602-412-3051
info@securosis.com www.securosis.com

Author's Note

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

Special thanks to Chris Pepper for editing and content support.

Thanks to our Sponsors for this project:

nCircle

nCircle is the leading provider of information risk and security performance management solutions to more than 6,500 businesses and government agencies worldwide. nCircle solutions enable enterprises of all sizes to (1) automate compliance and reduce risk, and (2) measure and compare the performance of their IT security program with their own goals and industry peers. nCircle solutions may be deployed on a customer's premises, as a cloud-based service, or in combination, for maximum flexibility and value.

nCircle has won numerous awards for growth, innovation, customer satisfaction and technology leadership and has been ranked among the top 100 best places to work in the San Francisco Bay Area. nCircle is headquartered in San Francisco, CA, with regional offices throughout the United States and in London and Toronto. To learn how you can more effectively protect your company visit us at http://www.ncircle.com.

Qualys

Qualys, Inc. is a pioneer and a leading provider of information security and compliance cloud solutions with 5,700+ customers in 85 countries, including 51 of the Forbes Global 100. The QualysGuard Cloud Platform and integrated suite of applications helps businesses simplify security operations and lower the cost of compliance by delivering critical security intelligence on demand and automating the full spectrum of auditing, compliance and protection for IT systems and web applications. Founded in 1999, Qualys has established strategic partnerships with leading managed service providers and consulting organizations including BT, Dell SecureWorks, Fujitsu, IBM, NTT, Symantec, Verizon, and Wipro. The company is also a founding member of the Cloud Security Alliance (CSA). For more information, please visit www.qualys.com.

Rapid 7

Rapid7 is the leading provider of security risk intelligence solutions. Rapid7's integrated vulnerability management and penetration testing products, Nexpose and Metasploit, empower organizations to obtain accurate, actionable and contextual intelligence into their threat and risk posture. Rapid7's solutions are being used by more than 2,000 enterprises and government agencies in more than 65 countries worldwide, while the Company's free products are downloaded more than one million times per year and enhanced further by over 125,000 security community users and contributors. Rapid7 has been recognized as one of the fastest growing security companies by Inc. Magazine and as a "Top Place to Work" by the Boston Globe. The Company is backed by Bain Capital Ventures and Technology Crossover Ventures. For more information about Rapid7, please visit http://www.rapid7.com.

Tenable Network Security

Tenable Network Security is the leader in Unified Security Monitoring. Tenable provides enterprise-class agentless solutions for the continuous monitoring of vulnerabilities, configuration weaknesses, data leakage, log management and compromise detection to help ensure FDCC, FISMA, SANS CAG and PCI compliance. Tenable's award-winning products are utilized by many Global 2000 organizations and Government agencies to proactively minimize network risk. For more information, please visit http://www.tenable.com/.

Contributors

The following individuals provided comments on the Securosis blog, which were factored into the paper.

Adrian Lane
Betsy Nichols
Eric Perraudeau, Qualys
Jeff LoSapio, Stratum Security

Copyright

Table of Contents

Introduction

Back when *The Pragmatic CSO* was published in 2007, I sent out a set of tips for being a better CISO as part of the promotional efforts. Tip #2 is *Prioritize Fiercely.* Let's take a look at this tip.

> *Tip #2 is all about the need to prioritize. The fact is you can't get everything done. Not by a long shot. So you have a choice. You can just not get to things and hope you don't end up overly exposed. Or you can think about what's important to your business and act to protect those systems first. Which do you think is the better approach?*
>
> *The fact is that any exposure can create problems. But you dramatically reduce the odds of a career-limiting incident if you focus most of your time on the highest profile systems. Maybe it's not good old Pareto's 80/20 rule, but you should be spending the bulk of your time focused on the systems that are most important to your business. Or hope the bad guys don't know which is which.*

5 years later that advice still makes perfect sense. No organization, including the biggest of the big, has enough resources. So you need to make tough choices. Things won't all be done when they need to be. Some things won't get done at all. So how do you choose?

No organization, including the biggest of the big, has enough resources. Which means you must make tough choices. So how do you choose?

Unfortunately most organizations don't choose at all. They do whatever is next on the list, without much rhyme or reason determining where things land on it. It's the path of least resistance for a tactically oriented environment. Oil the squeakiest wheel. Keep your job. It's all very understandable, but not very effective.

Optimally, resources are allocated and priorities set based on their value to the business. In a security context, that means the next thing done should reduce the most risk to your organization. Of course calculating that *risk* is where things get sticky. Regardless of your specific risk quantification religion, we can all agree that you need data to accurately evaluate the risks and prioritize. Last year we did a research project called Fact-Based Network Security: Metrics and the Pursuit of Prioritization which dealt with one aspect of this problem: how to make decisions based on network metrics.

But the issue is bigger than that. Network exposure is only one factor in the decision-making process. You need to factor in a lot of other data — including vulnerability scans, device configurations, attack paths, application and database postures, security intelligence, benchmarks, and lots of other stuff — to get a full view of the environment, evaluate the risk, and make appropriate prioritization decisions. Historically, vulnerability scanners have provided a piece of that data, telling you which devices were vulnerable to what attacks. The scanners didn't tell you whether the devices were really *at risk* — only whether they were vulnerable.

From Tactical to Strategic

Organizations have traditionally viewed vulnerability scanners as a tactical product, largely commoditized, and only providing value around audit time. How useful is a 100-page vulnerability report to an operations person trying to figure out what to fix next? Though those 100-page reports make auditors smile, as they offer a nice listing of all the audit deficiencies to address in the findings of fact.

The tide is definitely turning. We see a clear shift from a largely compliance-driven orientation to a more security-centric view. It's widely acknowledged that compliance provides a low (okay — *very* low) bar for security, and it just isn't high enough. So more strategic security organizations need better optics. They need the ability to analyze threat-related data, combine it with an understanding of what is vulnerable, and provide visibility to what is meaningfully **at risk**.

Yesterday's vulnerability scanners are evolving to meet this need, and emerging as a much more strategic component of an organization's control set than in the past.

Yesterday's vulnerability scanners are evolving to meet this need, and emerging as a much more strategic component of an organization's control set than in the past. As with last year's SIEM Replacement research, we believe it is now time to revisit your threat management/vulnerability scanning strategy. Not necessarily to swap out products, services, or vendors — but to ensure your capabilities map to what you need now and in the future. We will start by covering the traditional scanning technologies and then quickly go on to some advanced capabilities you will need to start leveraging these platforms for decision support. Yes, "decision support" is the fancy term for *helping you prioritize.*

Platform Emergence

You need more than just a set of tactical scans to generate a huge list of things you'll never get to. You need information that helps you decide how to allocate resources and prioritize efforts. We believe what used to be called a "vulnerability scanner" is evolving into a *vulnerability/threat management platform*. Sounds spiffy, right?

When someone says *platform* that usually indicates use of a common data model as the foundation, with a number of different applications riding on top to deliver value to customers. You don't *buy* a platform *per se*. You buy applications that leverage a platform to provide value to solve the problems you have. That's exactly what we are talking about here. But traditional scanning technology isn't a platform in any way. So this vulnerability management evolution requires a definite technology evolution. We are talking about growth from a single-purpose product into a multi-function platform.

This evolved platform encompasses a number of different capabilities. Starting with the tried and true device scanner, adding database and application scanning and risk scoring. We will describe not just the core technology that enables the platform, but the critical enterprise integration points and bundled value-added technologies (including attack path analysis, automated pen testing, and benchmarking) that distinguish a strategic platform from a tactical product. We will also describe the enterprise features you need from a platform, including workflows and dashboards, to complete the picture.

Scanning the Infrastructure

Traditional vulnerability scanners, focused purely on infrastructure devices, do not provide enough context to help organizations prioritize their efforts. Those traditional scanners are the plumbing of threat management. You don't appreciate the scanner until your proverbial toilet is overflowing with attackers and you have neither a mop nor any idea what are they targeting. The bulk of this research focuses on the case for transcending device scanning, but it remains a core component of any evolved vulnerability/threat management platform. So let's look at some key aspects of a traditional scanner.

Core Features

As implementations of a mature technology, pretty much all the commercial scanners have a core set of functions that work well. Of course different scanners have different strengths and weaknesses, but for the most part they all do the following:

You don't appreciate the scanner until your proverbial toilet is overflowing with attackers and you have neither a mop nor any idea what are they targeting.

- **Discovery:** You can't protect something (or know it's vulnerable) if you don't know it exists. So the first key feature is discovery. The enemy of a security professional is surprise, so make sure you know about new devices as quickly as possible, including rogue wireless access points and other mobile devices. Given the need to discover continuously, passive scanning and/or network flow analysis can be interesting and useful complements to active device discovery.
- **Device/Protocol Support:** Once you find a device you need to determine its security posture. Compliance demands that we scan all devices with access to private/sensitive/protected data, so any scanner should assess the varieties of network and security devices running in your environment, as well as servers running all relevant operating systems. Of course, databases and applications are important too, so we'll discuss those later. And be careful scanning brittle systems like SCADA — knocking down production devices doesn't make any friends in Ops.
- **Inside out and Outside in:** You can't assume adversaries are purely external (or internal), so you need the ability to assess devices from both inside and outside your network. Some kind of scanner appliance (which could be virtualized) is needed to scan the innards of your environment. You'll also want to monitor your IP space from the outside to identify new Internet-facing devices, find open ports, etc.

- **Accuracy:** Unless you enjoy wild goose chases you'll appreciate a scanner that minimizes false positives by focusing on accuracy.
- **Accessible Vulnerability Information:** Every found vulnerability requires a determination of severity, so it's very helpful to have information — from either the vendor's research team or other third parties — on the vulnerability directly within the scanning console.
- **Appropriate Scale:** Adding capabilities to the evolved platform makes scale a much more serious issue. But first the scanner must be able to scan your environment quickly and effectively — whether that is 200 or 200,000 devices. The point is to ensure the scanner is extensible enough to what you need as you add devices, databases, apps, virtual instances, etc. over time. We will discuss platform architectures later, but for the moment suffice it to say there will be a lot more data in the vulnerability management platform, and the underlying platform architecture needs to keep up.
- **New and Updated Tests:** Organizations face new attacks constantly, and they never stop evolving. So your scanner needs to stay current to test for the latest attacks. Exploit code based on patches and public vulnerability disclosures typically appears within a day so time is of the essence. Expect your platform provider to make significant investments in research to track new vulnerabilities, attacks, and exploits. Scanners need to be updated almost daily, so you need the ability to update them with new tests transparently — whether on-premises or in the cloud.

Additional Capabilities

But that's not all. Today's infrastructure scanners also offer what were formerly value-added functions that have become increasingly critical and now need to be included in the base platform. These include:

- **Configuration Assessment:** There really shouldn't be a distinction between scanning for a vulnerability and checking for a bad configuration. Either situation provide an opportunity for compromise. For example, a patched firewall with an any-to-any policy doesn't protect much — regardless of any vulnerability defects. But unfortunately the industry's focus on vulnerabilities means this capability is usually considered a scanner add-on. It shouldn't be. We expect both vulnerability scanning and configuration assessment to emerge as critical platform components. Further evolution will add the ability to monitor for system file changes and integrity — it is the same underlying technology.
- **Policy Management:** The vulnerability/threat management platform is driven by polices. When to scan, what to scan, which configuration policies are in force, which patches have been vetted and authorized and should now be applied. At times customer policies may be at odds with vendor recommendations, so the policies need to be flexible enough to assess whatever you decide needs to be assessed.
- **Patch Validation:** As described in Patch Management Quant, validating patches is an integral part of the process. With some strategic integration between patch and configuration management, the vulnerability/threat management platform can (and should) verify installed patches to confirm that the vulnerability has been remediated. Further integration involves leveraging workflow to automate the sending information to and from Ops systems to close the loop between Security and Ops.

Cloud and Virtualization Support

With the accelerating adoption of virtualization in data centers, the boundaries of what we consider the *data center* have shifted, along with the meanings of **in** and **out**. That means … you guessed it … your vulnerability/threat management tactics and techniques need to change.

Support for "cloud computing" means different things to different people. Let's narrow it down.

- Factor in the rapid addition and removal of virtual machines. This means not only assessing hypervisors as part of your attack surface, but also integrating information from the virtualization management console (vCenter, etc.) to discover which devices are in use and which are not.
- Verify the information coming from the virtualization console. You learned not to trust anything in security preschool, didn't you?
- Make sure your platform has sufficient API support. The main integration point with cloud computing platforms is the APIs offered by the IaaS (Infrastructure as a Service) and virtualization infrastructure providers.

So what's the difference between all these capabilities and what you already have? It's all about making 1 + 1 = 3 by integrating data to derive information and drive priorities.

Leveraging Collection

So what's the difference between all these capabilities and what you already have? It's all about making 1 + 1 = 3 by integrating data to derive information and drive priorities. We have seen some value-added capabilities (configuration assessment, patch validation, etc.) further integrated into infrastructure scanners to good effect. This positions the vulnerability/threat management platform as another source of intelligence for security professionals.

Scanning the Application Layer

Our research indicates most attacks target applications directly, so we can no longer just scan the infrastructure and be done with it. We need to *climb the stack* and pay attention to the application layer, looking for vulnerabilities in applications as well as their supporting components. But that requires us to define an 'application', which is surprisingly difficult.

A few years ago the definition of *application* was fairly straightforward. Even in an N-tier app, with a variety of application servers and data stores, you largely controlled all the components of the application. Nowadays, not so much. Pre-assembled web stacks, open source application servers, third-party crypto libraries, and cloud services all make for quick application development, but blur the line between your application and its supporting infrastructure. You have little visibility into what's going on behind the curtain but you're still responsible for securing it. For our vulnerability/threat management discussion we define the app as *presentation* and *infrastructure*. The presentation layer focuses on assembling information from a number of different sources — whether internal or external to your enterprise. The user of the application couldn't care less where the data comes from. So you need to assess the presentation code for issues that put devices at risk.

Pre-assembled web stacks, open source application servers, third-party crypto libraries, and cloud services all make for quick application development, but blur the line between your application and its supporting infrastructure.

But your focus on reducing the attack surface of applications also requires you to pay attention to the infrastructure. That means application servers, interfaces, and databases that assemble the data presented by the application. So you scan application servers and databases to find problems. Let's dig into these two aspects of the application to assess: databases and application infrastructure.

Database Layer

Assessing databases is more like scanning infrastructure than applications — you look for vulnerabilities in the DBMS (database management system). As with other infrastructure devices, databases can be misconfigured and might have improper entitlements, all of which pose risks. So assessment needs to consider whether appropriate database patches have been installed, the configuration of the database, improper access control, entitlements, etc…

Let's work through the key steps in database assessment:

- **Discovery:** First you need to know where your databases are. That requires a discovery process, preferably automated to find both known and unknown databases. You need to be wary of shadow IT, where lines of business and other groups build their own data stores — perhaps without the operational mojo of your data center group. You should also ensure you are continuously searching for new databases because they pop up anywhere at any time, just like rogue access points.

- **Vulnerabilities:** You will also look for vulnerabilities in your DBMS platform, which requires up-to-date tests for database issues. Your DB assessment provider should have a research team to keep track of the newest and latest attacks on whatever database platforms you use. Once something is found information about exposure, workarounds, and remediations is critical for making your job easier.

- **Configurations:** Configuration checking a DBMS is slightly different — you assess mostly internals. Be sure to check the database both with authorized user credentials and without (to better represent a typical outside attacker). Both scenarios are common in database attacks, so make sure your configuration is tight against both of them.

- **Access Rights and Entitlements:** Aside from default accounts and passwords, focus your efforts on making sure no users (neither human nor application) have inappropriate entitlements that put the database platform at risk. For example you need to ensure credentials of de-provisioned users have been removed and that accounts which only need read access don't have the ability to `DROP TABLES`. And you need to verify that users — especially administrators — cannot 'backdoor' the database through local system privileges. This is housekeeping to a degree but you still need to pay attention — make sure your databases are configured correctly to avoid unnecessary risk.

Finally, we have focused more on vulnerability/threat identification and assessment, but over time you will see even tighter integration between evolved vulnerability/threat management platforms and tactics to remediate problems. We wrote a detailed research report on Database Assessment, and you should track our Database Security Platform research closely to shorten your exposure window by catching problems and taking action more quickly.

Application Layer

Application assessment (especially of web applications) is a different animal. Mostly because you have to actually 'attack' the application to find vulnerabilities, which might exist within the application code or the infrastructure components it is built on. Obviously you need to crawl through the app to find and fix issues. There are several different types of app security testing (as discussed in Building a Web App Security Program), so we will just summarize here.

- **Platform Vulnerabilities:** This is the stuff we check for when scanning infrastructure and databases. Applications aren't 'stand-alone' — they depend on infrastructure and inherit vulnerabilities from their underlying components. The best example is a content management system, where a web app built on Drupal inherits all the vulnerabilities of Drupal, unless they are somehow patched or worked around.

- **Static Application Security Testing (SAST):** Also called "white box testing", SAST involves developers analyzing source to identify coding errors. This is not normally handled by security teams — it is normally part of a secure development lifecycle (SDLC).

- **Dynamic Application Security Testing (DAST):** Also known as "black box testing", DAST is an attempt to find application defects using bad inputs, using fuzzing and other techniques. This doesn't require access to the source code, so some security teams get involved in DAST, but it is still largely seen as a development responsibility because thorough DAST testing can be destructive to the app, and so shouldn't be used on production applications.

Web App Scanners

The technology most relevant to the evolution of vulnerability management is the web application scanner. Many of the available vulnerability management offerings offer an add-on capability to scan applications and their underlying infrastructures to identify vulnerabilities by automating the types of attacks typically used by web attackers. So what's the difference between a web app scanner and DAST? Mostly depth of analysis. Web app scanning can (and should) happen both before and after deployment, and tends to be the responsibility of the security and/or audit team. Over time we expect the features and functions of web app scanners and DAST tools to overlap more and more.

The key capabilities of a web application scanner are: 1) to discover the web applications in use in the enterprise, and 2) to automate testing of applications against common attacks.

> Any automated tool inevitably generates a bunch of false positives, and every alert needs to be investigated by a human to determine whether it represents a real issue.

In terms of discovery, as mentioned above for infrastructure devices, you can't assess what you don't know about. So populating and maintaining this inventory of web applications becomes critical. It's not like business folks and the developers remember to tell security about every new application as a matter of course.

In terms of automation, we are thinking of the typical attacks launched against web applications — such as cross-site scripting (XSS), SQL injection, and directory traversal. Most of the web app scanners available today offer from 25 to 40 distinct attacks to test. You will also see a lot of verbiage about supporting attack lists such as the OWASP Top 10 and the SANS 20 Critical Security Controls. Make sure the tool you select can perform a comprehensive set of attacks against your applications.

But a number of the attacks in those lists fall outside the purview of any automated scanner because they target application logic. Keep the generic nature of those lists in mind as you use them, as well as the inherent limitations of any tool launching automated web attacks. Your tool should track known vulnerabilities in common application platforms, from PHP through full content management and blogging systems such as Drupal and WordPress.

The other key feature to consider is accuracy. Applications are complicated beasts, typically with a number of controls in play at any given time. So any automated tool inevitably generates a bunch of false positives, and every alert needs to be investigated by a human to determine whether it represents a real issue. Part of your evaluation process should involve using the scanner against some existing applications to evaluate the number and type of false positives produced. You can't avoid them completely, but you can minimize the impact and look for a better signal:noise ratio. Don't buy a tool that creates more work than it eliminates.

Finally, keep in mind the importance of integration at all layers of the assessment. Developers can and should leverage any information they get to improve and make sure their applications are as secure as possible. So sending information from the web scanning tool to the developers' DAST environment can help narrow down issues sooner and fix those issues more efficiently.

Human Factors

There are limitations to how much DAST or a Web App Scanner can find. There is no way for an automated technology to detect logic flaws within an application. To truly understand how an application can be compromised you also need a human to check its logic, respond to its error codes, and generally validate whether found vulnerabilities present actual exposure. We are big fans of tools doing the grunt work and providing a level of code coverage not available or affordable using manual techniques. But you always need a skilled analyst to wade through the results and understand what is really an exposure that needs to be dealt with immediately.

Core Technologies

So far we have focused mostly on tactical stuff. Run the scan, get a report, fix stuff (or not), and move on. But for a *strategic and evolved* threat management platform emerging from its vulnerability scanning heritage, the core technology needs to serve more than merely tactical goals — it must provide a foundation for a number of additional capabilities. Let's review the key requirements. You need the ability to to scan/assess:

1. **Critical Assets:** This includes the key elements in your critical data path — it requires both scanning and configuration assessment/policy checking for applications, databases, server and network devices, etc.

2. **Scale:** Scalability requirements reflect the needs of your specific environment, not a generic idea of how much you can theoretically do with the platform. You want to be sure the platform's deployment architecture will support all of the devices you need assessed, deliver the analytics you need in a timely fashion without consuming all your network bandwidth.

3. **Accuracy:** You don't have time to mess around, and you don't want a report with 1,000 vulnerabilities, 400 of them false positives. There is no way to totally avoid false positives (aside from not scanning at all) so accuracy is a key selection criteria.

4. **Non-destructive behavior:** It's important to ensure scans are not destructive. You don't want to knock down a brittle control system with a scan that locks it up. Obviously there is a time and place for tests using exploit code (as we will describe later), but you need to make sure your platform can be configured to run only appropriate tests on its targets.

Yes, that's pretty obvious. With a mature technology like vulnerability management the question is less what you need to do and more how — especially when positioning for evolution and advanced capabilities. So let's dig into the foundation of any kind of strategic platform: the data model.

Integrated Data Model

What's the difference between a tactical scanner and an integrated vulnerability/threat management platform? Data sharing, of course. The platform needs to consume and store more than just scan results. You also need configuration data, third party and internal research on vulnerabilities, assessment of attack paths, and a bunch of other data types we will discuss when we get into value-add technology. Flexibility and extensibility are key for the data schema. Don't get stuck with a rigid schema that won't allow you to add whatever data you need to most effectively prioritize your efforts — whatever data that may turn out to be.

What's the difference between a tactical scanner and an integrated vulnerability/threat management platform? Data sharing, of course.

Once the data is in the foundation the next requirements involve analytics. You need to set alerts and thresholds on the data and correlate disparate information sources to glean perspective and support decision-making. We are focused on more effectively prioritizing security team efforts, so your platform needs analytical capabilities to help turn all that data into useful information.

When you start evaluating specific vendor offerings you may get dragged into a discussion of storage approaches and technologies. Does a relational backend, or an object store, or even a proprietary flat file system provide the performance and flexibility to serve as the foundation of your platform. Or even whether a general-purpose business intelligence engine is more appropriate for the types and volume of data you need to analyze. Understand that it really is a religious discussion, and the vendors are evangelists will try to convince you that their way is the only path to salvation. But your analysis efforts need to focus on the scale and flexibility of *whatever* data model underlies the platform.

Also pay attention to evolution and migration strategies, especially if you plan to stick with your current vendor as they move to a new platform. A transition to a new data model is something like a brain transplant, so make sure the vendor has a clear and well-thought-out path. Obviously if your vendor stores their data in the cloud it's not your problem, but we will discuss the cloud versus customer premises later.

Discovery

Once you get to platform capabilities you start by determining what's in your environment. That means a *discovery* process to find devices on your network and make sure everything is accounted for. Avoid the "oh crap" moment, when a bunch of unknown devices and applications show up — and you have no idea what they are, what they have access to, or whether they are steaming piles of malware. At least shorten the window between something showing up on your network and discovering it.

> Avoid the "oh crap" moment, when a bunch of unknown devices and applications show up — and you have no idea what they are, what they have access to, or whether they are steaming piles of malware.

There are a number of techniques for discovery, including actively scanning your entire address space for devices and profiling what you find. That works well enough and is traditionally the main way vulnerability management offerings handle discovery, so active discovery is still table stakes for vulnerability scanners. You need to balance the network impact of active discovery against the need to quickly find new devices. And make sure you can search your networks completely, which means both your IPv4 space and your emerging IPv6 environment. Oh, you don't have IPv6? Think again. You'd be surprised at the number of devices that ship with IPv6 active by default, and if you don't plan to discover that address space as well you'll miss a significant attack surface.

You can supplement active discovery with a passive capability that monitors network traffic and identifies new devices based on network communications. Depending on the sophistication of the passive analysis, devices can be profiled and vulnerabilities can be identified, but the primary goal of passive monitoring is to find new unmanaged devices faster. Once a new device is identified passively you can launch an active scan to figure out what it's doing or kick off a workflow to make sure that belongs. Passive discovery is also helpful for identifying devices hidden behind firewalls and on protected segments, which block active discovery and vulnerability scanning.

But that's not all — depending on the breadth of your vulnerability/threat management program you might want to discover endpoints and mobile devices. We always want more data, so we are partial to discovering *all* assets in your environment. That said, in terms of determining what's important in your environment (see the asset management/risk scoring section below), endpoints tend to be less important than databases with protected data, so prioritize your effort on discovery and assessment.

Finally, another complicating factor for discovery is cloud computing. With the ability to spin up and take down instances at — perhaps outside your data center — your platform needs to both track and assess cloud resources, which requires integrating with cloud consoles to ensure your platform knows about new devices and can assess them appropriately. This is an emerging capability, but realistically we will see a lot more private and public cloud resources everywhere.

Asset Management and Risk Scoring

Remember the key capability of the evolved vulnerability/threat management platform is its ability to help you prioritize. So any *calculation* of a risk score largely depends on 1) the 'importance' of the asset and 2) how 'exposed' it is to attack at any given point in time. Evaluating what's important begins as an asset management function. Of course many operations teams already have extensive asset management efforts, so the platform should integrate with and take advantage of any existing resources. But many organizations don't have an asset database (scary as that sounds), so the platform may need to serve as the authoritative registry of IT assets. Either way, the platform needs to store and/or access asset information. And don't forget that cloud computing is changing the definition of 'asset', so all the concepts we defined above (in Scanning the Infrastructure) apply here as well.

The more flexible the system the better; your platform should support the way **you** categorize assets — not force you to fit your assets into the vendor's buckets.

Once you have the assets defined the next step is to tag, group, and categorize them. The more flexible the system the better; your platform should support the way **you** categorize assets — not force you to fit your assets into their vendor's buckets. Assign an (admittedly subjective) importance to each group or category of assets. We suggest a simple approach, with 3 or 5 levels of importance. Really important means someone would be fired, but other assets are simply not important. You don't need complexity or fine precision, but you do need to identify devices which hold or have access to critical data.

As you evaluate the vulnerability of each asset through the platform's various tests you can determine a risk score for prioritization.

The key is flexibility. Group assets in a way that makes sense for your organization. Derive a risk score based on your calculation of risk, not an black box that may or may not be appropriate for your organization. And you need the ability to change everything the next time a significant technology or organizational disruption happens, like cloud computing or a big M&A deal.

To Cloud or Not to Cloud

Now let's address the *cloud* buzzword. In this context 'cloud' means SaaS (software as a service), and the vendor manages the infrastructure, which you access via a browser interface across the Internet. If you thought people got religious about data models and engines, ask a cloud vendor about an on-premise solution or vice-versa. That's always fun. At the end of the day, this cloud discussion involves two things:

1. **Scale:** You will hear a lot from cloud providers about infinite scale and the limitations of customer premise offerings. It is true that scalability is the vendor's problem in a cloud scenario. That offers some advantages, but any solution can scale with a suitable deployment architecture.

2. **Technology Updates/Change:** The other big message you'll hear from cloud bigots is that cloud platforms handle software updates more quickly and transparently than gear that runs onsite. Again, there is truth here, but every vulnerability management vendor has been sending new rules and tests to its devices for years, so it's not like they haven't figured out software distribution and updating.

The 'decision' isn't really a decision at all — what is and isn't 'cloud' nowadays is largely a matter of semantics.

These two objections to customer premise solutions are really much ado about nothing. The 'decision' isn't really a decision at all — what is and isn't 'cloud' nowadays is largely a matter of semantics. Let's get back to *your* requirements. You need to be able to test your environment from outside — most attackers are outside your perimeter — which works best with a cloud service. But you also need a presence within your perimeter to scan internal devices, especially those on protected networks. So every cloud service must include an on-site component for internal scans.

The on-site component might be a dedicated appliance, a virtual machine, a dynamic instance downloaded to a device inside the network at scan time, or some combination. There is no point in getting religious about deployment models, and all the leading platform vendors offer hybrid approaches to meet your specific needs. If they don't they probably aren't right for your environment. But don't get caught up in hype. You need an external component to test your environment from the outside and an internal component to test inside your perimeter.

Agent or None? With or without Credentials?

You will hear a lot about agent vs. agent-less scanning. This is also mostly hyperbole and semantics. In order to do any kind of granular scanning of a device, you need a persistent agent on the device, the ability to download a temporary agent, or full administrator rights (credentials) to the device to remotely poll it for the things you are looking for (configurations, patches, logs, etc.). As usual, the correct answer is *all of the above*. There are advantages to a temporary agent in terms of simpler software distribution to devices you worry about. But ultimately the scanning model you choose depends on your access to the device, the type of device, and what kinds of data it has access to. Obviously you don't want to introduce more problems than you solve by adding an agent, so pay attention to the stability and manageability of any software you require on endpoints and servers.

The correct answer to "Credentialed or non-credentialed scans?" is also *both*. Non-credentialed scans give you the external attacker's view, but they of course provide less detail than a credentialed scan. So a full understanding of a device's security posture also requires a credentialed scan with full access to configurations, patch levels, logs, entitlements, applications, etc.

Keep in mind that you *cannot* actively scan certain devices, such as brittle control systems which fall over under the onslaught of a vulnerability scan. Above we mentioned passively discovering assets by monitoring the network. A similar approach can find vulnerabilities on devices you can't actively scan. Obviously it provides less detail than a credentialed scan, but i's better to get *some* data without knocking down the device.

Security Research

Finally, any vulnerability/threat management platform needs to be driven by research. Things move fast in the attack space and your threat management tools need to stay current. So your vendor needs to make a considerable investment in a dedicated team to track the field, observe and analyze new attacks, figure out how to search for those attacks using their tools, ensure the quality of their tests to minimize false positives, and finally get the tests into your hands as quickly as possible. For a more granular view into the process of analyzing attacks and malware, check out the Analyze Malware subprocess in Malware Analysis Quant. It provides an idea of what's involved in profiling malware files and figuring out how to find them.

Things move fast in the attack space and your threat management tools need to stay current.

To compare research groups evaluate the sophistication of their analysis. Do you understand how to remediate issues your scanner finds? Can you determine the seriousness of the attack? Do you believe them? Is the data just coming from one vendor or do they integrate third party data? And most importantly, do they provide coverage for your assets? You know, the operating systems, databases, and critical applications that drive your business.

Value-Add Technologies

So far we have talked about scanning infrastructure and the application layer, as well as some technology decisions such as how to deal with cloud/SaaS delivery and agents. But as much as these capabilities may increase the value of the vulnerability/threat management platform, it's still not enough to prioritize the hundreds (if not thousands) of vulnerabilities or configuration problems you'll find to really focus security efforts. Let's look at a few emerging capabilities that help make the information gleaned from scans and assessment more relevant to the operational decisions you make every day.

These capabilities are not common to all the current leading vulnerability management offerings. But we expect most or all to be core capabilities of these platforms in some way over the next 2-3 years, so watch for increasing M&A and technology integration for these functions.

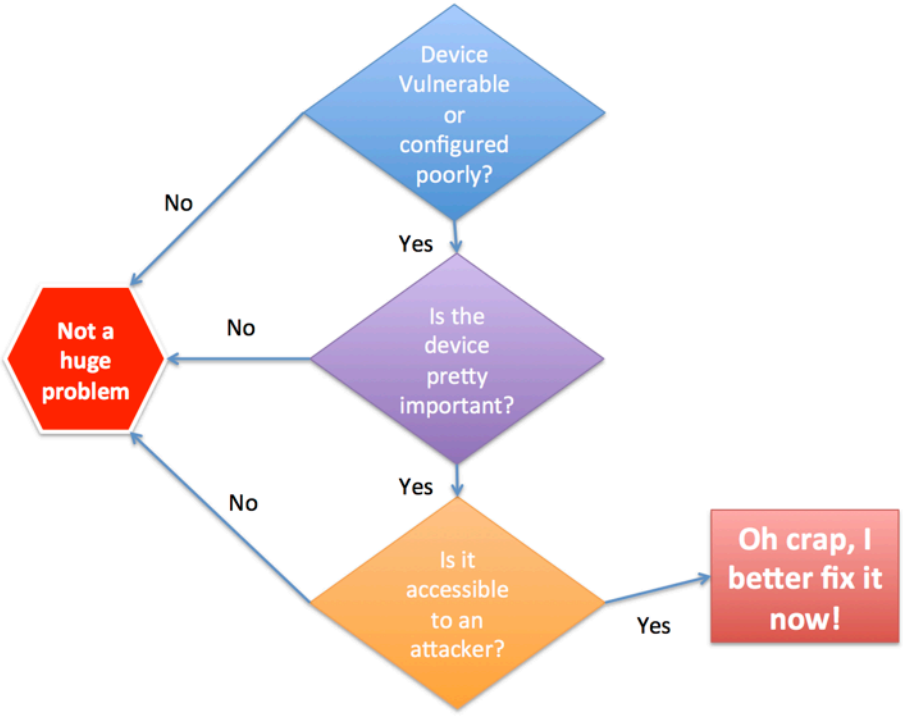
These capabilities are not common to all the current leading vulnerability management offerings. But we expect most or all to be core capabilities of these platforms in some way over the next 2-3 years.

Attack Path Analysis

If no one hears a tree fall in the woods, has it really fallen? The same can be asked about a vulnerable system. If an attacker can't get to the vulnerable device, is it really vulnerable? The answer is yes, it's still vulnerable, but remediation is less urgent. So tracking which assets are accessible to a variety of potential attackers is critical for an evolved vulnerability management platform.

This analysis is typically based on ingesting firewall rule sets and router/switch configuration files. With some advanced analytics the tool determines whether an attacker could (theoretically) reach the vulnerable devices. This adds a critical third leg to the "Oh crap, I better fix it now!" decision process depicted below.

Obviously most enterprises have complicated networks, which means an attack path analysis tool must be able to crunch a *huge* amount of data to work through all the permutations and combinations of possible paths to each asset. You should also look for native support of your devices (firewalls, routers, switches, etc.) so you don't have to do a bunch of manual data entry — given the frequency of change in most environments, this is likely to be a complete non-starter. Finally, make sure the visualization and reports on paths present the information in a way you can use.



By the way, attack path analysis tools are not really new. They have existed for a long time but never really achieved broad market adoption. We are big fans of Mr. Market, so we need to ask what's different now that would enable the market to develop? First, integration with the vulnerability/threat management platforms makes this information part of the threat management cycle rather than a stand-alone function, and that's essential. Second, modern tools can finally offer analysis and visualization at enterprise scale. We expect this technology to be a key part of the platforms sooner rather than later — we already see some early technical integration deals, and expect more.

Automated Penetration Testing

Another key question raised by a long vulnerability report is, "Can you exploit this vulnerability?" Like a vulnerable asset without an open attack path, if a vulnerability cannot be exploited due to some other control or the lack of a weaponized exploit remediation is less urgent. For example, a HIPS deployed on a sensitive server might block attacks against a known vulnerability. Obviously a basic vulnerability scanner cannot detect that, so it will report the vulnerability just as urgently all your exploitable vulnerabilities.

The ability to actually run exploits against vulnerable devices as part of a security assurance process can provide perspective on what is really *at risk,* and not just *theoretically* vulnerable. In an integrated scenario a discovered vulnerability can be tested for exploit immediately, to either shorten the window of exposure or provide immediate reassurance.

Of course there is risk with this approach, including the possibility of taking down production devices, so use pen testing tools with care. But to really know what can be exploited and what can't, you *need* live ammunition. And be sure to use fully vetted, professionally supported exploit code. You should have a real quality assurance process behind the exploits you try. It's cool to have an open source exploit, and on a test/analysis network using less stable code that's fine. But you probably don't want to launch an untested exploit against a production system. Not if you like your job, anyway.

Compliance Automation

In the rush to get back to our security roots, many folks have forgotten that the auditor is still going to show up every month/quarter/year and we need to be ready. Preparing for an audit burns resources that could otherwise be used on more strategic efforts, and is a necessary evil but you want to minimize the resources you spend on such a tactical function. Vulnerability scanning is a critical part of every compliance mandate, so scanners have pumped out PCI, HIPAA, SOX, NERC/FERC, etc. reports for years. But that's only the first step in compliance automation. Auditors need plenty of other data to determine whether your control set is sufficient to satisfy regulations. That includes things like configuration files, log records, and self-assessment questionnaires.

We expect to see increasingly robust compliance automation in these platforms. That means a workflow engine to help get ready for your assessment and a flexible integration model to allow storage of additional unstructured data in the system. The goal is to ensure that when the auditor shows up, your folks have already collected all the data they need and can easily produce it. The easier that is, the sooner the auditor will go away and let your folks get back to work.

You are already finding what's wrong (either vulnerable or improperly configured), so why not just fix it immediately?

Patch/Configuration Management

Finally, it's no stretch to see the value of broader configuration and/or patch management capabilities within the vulnerability/threat management platform. You are already finding what's wrong (either vulnerable or improperly configured), so why not just fix it immediately? There is plenty of overlap with existing configuration and patching tools, and you could just as easily make the case that those tools can and should add vulnerability management functions. Regardless of which *platform* ultimately handles vulnerability and patching/configuration, there are clear advantages to a common environment.

There are also clear disadvantages to integrating finding and fixing these issues, mostly concerning separation of duties. Your auditors may have an issue with the same platform being used to figure out what's wrong and then to fix it, but that's a point of discussion for your next assessment. Clearly, in terms of leverage and efficiency, the ability to find and fix problems in a single environment is attractive to understaffed shops (everyone) and folks who need to improve efficiency (also everyone).

Benchmarking

We are big advocates of benchmarking your environment against other organizations of similar size and industry to figure out where you stand. In fact, we did a research project on this last year. We find great value in comparing configurations / patch windows / malware infections / any other metrics against other companies, and this is another valuable data point for determining which actions need to be taken right now.

Exfiltration Analysis

The flip side to figuring out the attack paths that can be used to access devices is whether an attacker could remove data from your environment after they do compromise a device. This is a very early market without commercial offerings to provide this analysis (yet). Instead organizations tend to analyze exfiltration as part of a penetration test. That said, we expect this category to emerge over the next two years as a very interesting value-add to the vulnerability/threat management platform.

We understand that some of these value-add capabilities already exist within existing IT management stacks, so a vulnerability/threat management platform should cleanly integrate with those existing functions. We will discuss the logical integration points next. But for now suffice it to say that smaller organizations (and enterprises acting like small organizations) may favor leveraging a single platform to provide a closed loop for finding and fixing problems without having to deal with multiple systems. Over time it is quite possible that vulnerability/threat management platforms will evolve into a broader IT ops platform — offering deeper asset management, performance management, those configuration/patch capabilities, and other more traditional IT Ops functions.

Over time it's quite possible that vulnerability/threat management platforms will evolve into a broader IT ops platform.

Will the evolved vulnerability management platforms ever really reach this point of being everything to everyone? Probably not — the natural order of the IT business means that only one or perhaps two of the current players are likely to remain stand-alone entities. The rest will likely be acquired by larger providers and integrated into the acquirer's stacks, or go away altogether.

Enterprise Features and Integration

After talking mostly about the transition from an audit-centric tactical tool to a much more strategic platform providing security decision support, it is now time to look critically at what's required to make the platform work in your enterprise. That means providing both built-in tools to help manage your vulnerability management program and supporting integration with existing security and IT management tools.

Whether you select a new platform or stay with your incumbent provider, as you add functionality you will need to play nicely in your existing sandbox.

It is rare to have an opportunity to start fresh in a green field. So whether you select a new platform or stay with your incumbent provider, as you add functionality you will need to play nicely in your existing sandbox.

Managing the Vulnerability/Threat Management Program

We have been around way too long to actually believe that any tool or toolset can ever entirely solve any problem, so our research tends to focus on implementing programs to address problems rather than selecting products. Vulnerability management is no different, so let's list what you need to actually manage the program internally.

You need some basic information before you can attempt any kind of prioritization. That has been the research focus to date. Taking tactical scans and configuration assessments of the infrastructure and application layers, then combining then with perceived asset value and value-add technologies, and running analytics to provide usable information. But the fun begins once you have an idea of what needs to be fixed, with relative priorities.

Dashboards

Given the rate of change in today's organizations wading through a 200-page vulnerability report or doing manual differential comparisons of configuration files isn't efficient or scalable. Add in cloud computing and everything is happening even faster, making automation essential to security operations. You need to take information and visualize it in ways that makes sense for a variety of constituencies. The first critical group to support are executives. You know, the folks who pay the bills. They should see a high-level view of current security posture and other important executive-level metrics.

The security team can leverage their own operational view for guidance on what to do. And you can probably use views for application-specific vulnerabilities, and perhaps infrastructure and database visuals for the application and database folks. You should have the flexibility to design an appropriate dashboard/interface for any staffer needing access to the platform's information. Most vendors ship with a bunch of out-of-the-box options, but more importantly offer a user-friendly capability to customize the interface for staff requirements.

An integral part of the workflow capability must be enforcement of proper separation of duties to ensure no one individual has too much control over your environment.

Workflow

Unless your IT shop is a one-man (or one-woman) band, some level of communication is required to keep everything straight. With a small enough team a daily coffee discussion might suffice. But that doesn't scale, so the vulnerability/threat management platform should include the ability to open 'tickets', or whatever you call them, to get work done. It certainly doesn't need to include a full-blown trouble ticket system, but this comes in handy if you don't have an existing support/help desk system. For basic functionality, look for the ability to do simple ticket routing, approve/authorize, and indicate work has been completed (close tickets). Obviously you will want extensive reporting on tickets, and an ability to give specific staff members to-do lists. Straightforward stuff.

Don't forget that any program needs to have checks and balances, so an integral part of the workflow capability must be enforcement of proper separation of duties to ensure no one individual has too much control over your environment. That means proper authorization before making changes or remediating issues, and a proper audit trail for everything administrators do with the platform.

Compliance Reporting

Finally you need to substantiate your controls for the inevitable audits, so your platform needs to generate documentation to satisfy the auditor's appetite for information. Okay, it won't totally satisfy the auditor (as if that were even possible) but should at least provide a good perspective on what you do and how well it works, with artifacts to prove it. Since most audits break down to some kind of *checklist* to follow, having those lists enumerated in the vulnerability management platform is important and saves a lot of time. You don't want to be mapping reports on firewall configurations to PCI Requirement 1 — the tool should do that out of the box. Make sure whatever you choose offers the reports you need for the mandates you are subject to.

But reporting shouldn't end when the auditor goes away. You should also use reports to keep everyone operationally honest, with reports on the same types of information as the dashboards mentioned above. You'll want senior folks to get periodic reports on open vulnerabilities and configuration problems, newly opened attack paths, and systems that can be exploited by the pen test tool. Similarly, operational folks might get reports of their overdue tasks or efficiency reports showing how quickly they remediate their assigned vulnerabilities. Again, look for customization — everyone seems to want information in their own format.

Dashboards and reporting are really the yin/yang of managing any security-oriented program. So make sure the platform provides flexibility to display and disseminate information however you need it.

Enterprise Integration

In today's technology environment nothing stands alone, so when looking at this evolved vulnerability management platform how well it integrates with what you already have is a strong consideration. But you have a lot of stuff, right? So let's prioritize that integration a bit.

- **Patch/Config Management:** We have already speculated a bit on the evolution of common platforms for vulnerability/threat and configuration/patch management. Tight integration between these two functions is clearly critical. You will probably hear the term *vulnerability validation* to describe this integration, which entails closing the loop between assessment and remediation. So when an issue is identified by the vulnerability/threat management platform, the fix is made (presumably by the patch/config tool), and then the platform verifies the fix was successful and eliminated the exposure. Checks and balances make for happy ops and security people.

- **Password Vaults:** A more granular scan of devices typically requires either an agent on the device or credentials to do a deep scan with administrator privileges. Most organizations do not want to load yet another agent onto devices, so credentialed scans are generally favored. That brings up the sticky issue of how to manage the credentials — especially for very sensitive devices in the data center. So having the vulnerability/threat management platform able to pull the credentials from a centrally managed password vault (described in detail in Watching the Watchers) can enable detailed scanning while maintaining the integrity and security of administrator credentials.

- **SIEM:** This is another area where integration makes sense. The data from the vulnerability/threat platform can help supplement log events, identity, network flow, and all the other data sources collected and analyzed by the SIEM. Of course there is overlap with the analytics from the evolved vulnerability/threat management platform, but that's not a bad thing — we like checks and balances.

- **Cloud Consoles:** The other area of integration is the emerging cloud computing management consoles. We talked about discovering devices on a continuous basis earlier, and integration with the cloud console helps identify new instances as they are spun up — in both private and public clouds. Getting information from the cloud console doesn't eliminate the need to use other traditional mechanisms to discover devices as well (trust but verify), but strategic integration provides a valuable head start. Of course the virtualization management players play API ping-pong and seem to constantly change their interfaces and integration points to maintain tighter control of the virtualization stack, but that's not your problem — it's the vulnerability management vendor's headache. You should expect a measure of integration with your cloud computing resources. Again, you can't assess it if you don't know it exists, especially in limited-visibility environments like public and private clouds.

You can't assess it if you don't know it exists, especially in limited-visibility environments like public and private clouds.

- **Pulling Data in:** As we have mentioned over and over again, we believe the vulnerability scanner is destined to evolve into a vulnerability/threat management platform. But calling a certain technology a "platform" intimates that it offers an infrastructure layer, basically an aggregation point for all sorts of data types, and "applications" that leverage the data to deliver certain functions to an organization. The evolution to a vulnerability/threat management platform means you'll need to import other enterprise threat data into the platform to provide this broader value proposition. So look for a robust API and flexible data model (described earlier under Core Technologies) to ensure rules, reports, and dashboards can be customized to leverage any additional data pulled into the platform.

Evolution or Revolution?

We have discussed the evolution of vulnerability management from a tactical tool into a much more strategic platform, providing decision support for folks to more effectively prioritize security operations and resource allocation. But some vendors may not manage to effectively broaden their platforms sufficiently to remain competitive and fully satisfy customer requirements. So at some point you may face a replacement decision, or to put it more kindly, a decision of *evolution or revolution* for your vulnerability/threat management platform.

You may face a replacement decision, or to put it more kindly, a decision of *evolution or revolution* for your vulnerability/threat management platform.

Last year we researched whether to replace your SIEM/Log Management platform. That research provides an in-depth process for revisiting your requirements, re-evaluating your existing tool, deciding whether to replace or not, negotiating the deal, and migrating to the new platform.

If and when you face a similar decision regarding your vulnerability/threat management platform the process will be largely the same, so check out that research for detail on the replacement process. The main difference is that, unlike SIEM platforms, most organizations are *not* totally unhappy with their current vulnerability tools. And again, in most cases a *revolution* decision results from a need to utilize additional capabilities available with a competing platform — not because the existing tool simply does not work (as is the case all too frequently with SIEM).

The Replacement Decision

Let's start with the obvious: you aren't really making a *decision* on the vulnerability management offering — it's more of a recommendation. The final decision will likely be made in the executive suite. That's why your process focuses initially on gathering data (quantitative when possible) — because you will need to defend your recommendation until the purchase order is signed. And probably afterwards — especially if a large 'strategic' vendor provides your current VM scanner.

This decision generally isn't about technical aspects — especially because there is an incumbent in play, possibly from a big company with important relationships with heavies in your shop. So to make any change you will need all your ducks in a row, and a compelling argument. And even then you might not be able to push through a full replacement, in which case the best answer may be to supplement.

In this scenario you still scan with the existing tool, but handle the value-add capabilities (web app scanning, attack path analysis, etc.) on the new platform. The replacement decision can be broken into a few discrete steps:

1. **Introspection:** Start by revisiting your requirements, both short and long term. Be particularly sensitive to how adversaries' tactics are changing. Unfortunately we haven't yet found a reliable crystal ball vendor, but think about how your infrastructure is and will be provisioned (cloud computing). What will your applications look like and who will manage them (SaaS)? How will you interact with business partners? Most important, be honest about what you really need. It's important to make a clear distinction between stuff you *must* have and things that would be *nice* to have. Everything looks shiny on a marketing spec sheet. That doesn't mean you'll really use those capabilities.

2. **Current Tool Assessment:** Does your current product meet your needs? Be careful to keep emotion out of your analysis — most folks get pissed at their existing vendors from time to time. Do some research into your current vendor's roadmap. Will they support the capabilities you need in the future? If so, when? Do you believe them? Don't be too skeptical, but if a vendor has a poor track record of shipping new functionality, factor that in.

3. **Alternatives and Substitutions:** You should be surveying the industry landscape to learn about other offerings that might meet your needs. It's okay to start gathering information from vendors — if a vendor can't convince you their platform will do what you need, they have no shot at actually solving your problem. But don't stop with vendors. Talk to other folks using the product. Talk to resellers and other third parties who can provide a more objective perspective on the technology. Do your due diligence, because if you push for a revolution it will be *your* fault if it doesn't meet expectations.

4. **Evaluate the Economics:** Now that you know which vendors could meet your requirements, what would it cost to get there? How much to buy the software, or is it a service? How does that compare to your current offering? What kind of concessions can you get from the new player to get in the door, and what will the incumbent do to keep your business? Don't make the mistake of only evaluating the acquisition cost — factor in training, integration, and support costs. And understand that you may need to run both offerings in parallel during a migration period to be sure you don't leave a gap in assessment.

5. **Document and Sell:** At this point your decision will be clear — at least to you. But you'll need to document what you want and why, especially if it involves bringing in another vendor. Depending on the political situation, consensus might be required among the folks affected by the decision. And don't be surprised by pushback if you decide on replacement. You never know who plays golf with whom, or what other big deals are on the table that could impact your project.

> Sometimes decisions don't go your way — no matter how airtight your case is.

And understand that you may not get what you want. It's an unfortunate reality of life in the big city. Sometimes decisions don't go your way — no matter how airtight your case is. That's why you are really only making a *recommendation*. Many different factors go into a replacement decision for a key technology, most of them beyond your control.

If your decision is to stay put and evolve the capabilities of your tool into the platform you need, then map out a plan to get there. When will you add the new features? Then you can map out your budgets and funding requests and work through the politics with your peers — VM platforms impact the network, security, and application teams.

But if your decision involves moving to another platform you will need significant planning and some savvy to successfully navigate the migration. So let's delve into some of those considerations in more detail.

Migration

Much of the complexity of migrating to a new vulnerability/threat management platform depends on how complicated your current environment is, which comes down to how much customization you have done. Have you built your own dashboards and reports? Have you scripted interactions with your help desk system? Are third-party data feeds integrated into the current tool? Do you send alerts and/or other data to an upstream aggregation point like a SIEM? If so those linkages need to be moved to the new offering, and even if everything goes well it still takes time — and might require additional investment and resources.

A flash cutover never really works for a tool you use every day. If you are only using the scanner for weekly or monthly scans that might work, though — you have a window to deploy and test the rules, and tune the dashboards and reports, before it's "go time" for the next set of scans.

But organizations which discover continuously and scan frequently need either an adequate window for cutover or a different approach. We recommend you start deploying the new vulnerability/threat management platform long before getting rid of the old. In the best case you can deprecate portions of the older system after replacement capabilities are online, but you will likely want the older system as a fallback until the new functions have been fully vetted. In our operational days we learned the importance of this staging process… the hard way. Ignore it at your peril — keeping in mind that your vulnerability/threat management platform underlies several key security and network operational functions.

We have broken the migration process into two phases: planning and implementation. Your plan needs to be very clear and specific about when things get installed, how data gets migrated, when you cut over from old systems to new, and who performs the work. Especially *who* — you won't get to forget about your other operational responsibilities while you migrate.

- **Planning:** First start the planning process by reviewing requirements and prioritizing the functions to implement. Focus your migration plan on getting some quick wins to build momentum early and demonstrate success. Once the major functions and associated milestones are defined you can allocate resources, define timelines, and prepare for the migration. By 'prepare' we mean: define scanning rules, revisit device configuration policies, aggregate topology information, design dashboards and reports, etc. Do as much work as possible ahead of the actual implementation, so once you start moving you can minimize recalibration during the process.
- **Implementation:** Once the plan is locked and loaded it's time to move. That involves deploying the (virtual) devices, installing policies, setting up dashboards and reports, testing and verifying functionality (false positives are especially annoying from a shiny new tool that was supposed to *solve* problems), getting acceptance from stakeholders, and finally decommissioning any other tools in use.

Once you have navigated the migration gauntlet you can kick back and enjoy the capabilities of your new platform to help prioritize efforts, improve efficiency, and generally increase the security of your environment. Or so you hope, anyway.

Summary

As we finish our analysis of the evolution of vulnerability scanners into vulnerability/threat management platforms, let's revisit the high points:

- **Stay focused on the prize:** It's critical to keep the objective of any vulnerability/threat management initiative in mind: *decision support* for security activities. Make sure that any data you collect in the platform helps prioritize your efforts based on facts, not intuition.
- **Climbing the Stack:** With a majority of attacks targeting applications and databases nowadays, vulnerability/threat management activities must include those layers in the analysis. That means web applications and databases become key targets for both scanning and configuration auditing. Attackers will look for any weakness in the system, so you can't ignore the network and devices, but much of the low hanging fruit is at the application layer.
- **The Path to Perdition:** You prioritize fixing specific issues based on whether the vulnerability of the device (either due to a known vulnerability or a configuration problem), the relative importance of the device (whatever 'important' means in your environment), and whether an attacker can actually reach the device. Attack path analysis is relatively new and one of the more exciting capabilities of these evolved vulnerability/threat management platforms.
- **A Cloudy Forecast:** Many folks would like you to believe that the deployment decision between an on-site and a cloud/SaaS platform is an either/or proposition. But the answer for most organizations is *both,* because there are critical discovery and scanning functions that need to happen both outside (for Internet-accessible devices) and inside (protected networks), so the decision really comes down to where you want your data stored, your analytics to run, and your interface hosted. There is no *right* or *wrong* here — it's a question of finding the right deployment model for your organization.
- **All Roads Lead to Home:** As the goal is prioritizing activities and more effectively allocating resources, there will be many paths to your desired goal. Perhaps you want to add configuration assessment to your scanning first. Or possibly pen test against vulnerable devices to get going. Perhaps analyzing the attack paths against critical devices is your initial focus. As long as you keep the end goal in mind — pulling all this data into a common platform for analytics — it's all good. There is no right or wrong path — just the one you are on.

If you have any questions on this topic, or want to discuss your situation specifically, feel free to send us a note at info@securosis.com or ask us a question via the Securosis Nexus <http://nexus.securosis.com>.

About the Analyst

Mike Rothman, Analyst/President

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

Starting his career as a programmer and networking consultant, Mike joined META Group in 1993 and spearheaded META's initial foray into information security research. Mike left META in 1998 to found SHYM Technology, a pioneer in the PKI software market, and then held executive roles at CipherTrust and TruSecure. After getting fed up with vendor life, Mike started Security Incite in 2006 to provide a voice of reason in an over-hyped yet underwhelming security industry. After taking a short detour as Senior VP, Strategy at eIQnetworks to chase shiny objects in security and compliance management, Mike joined Securosis with a rejuvenated cynicism about the state of security and what it takes to survive as a security professional.

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

About Securosis

Securosis, LLC is an independent research and analysis firm dedicated to thought leadership, objectivity, and transparency. Our analysts have all held executive level positions and are dedicated to providing high-value, pragmatic advisory services.

Our services include:

- **The Securosis Nexus:** The Securosis Nexus is an online environment to help you get your job done better and faster. It provides pragmatic research on security topics that tells you exactly what you need to know, backed with industry-leading expert advice to answer your questions. The Nexus was designed to be fast and easy to use, and to get you the information you need as quickly as possible. Access it at <https://nexus.securosis.com>.

- **Primary research publishing:** We currently release the vast majority of our research for free through our blog, and archive it in our Research Library. Most of these research documents can be sponsored for distribution on an annual basis. All published materials and presentations meet our strict objectivity requirements and conform to our Totally Transparent Research policy.

- **Research products and strategic advisory services for end users:** Securosis will be introducing a line of research products and inquiry-based subscription services designed to assist end user organizations in accelerating project and program success. Additional advisory projects are also available, including product selection assistance, technology and architecture strategy, education, security management evaluations, and risk assessment.

- **Retainer services for vendors:** Although we will accept briefings from anyone, some vendors opt for a tighter, ongoing relationship. We offer a number of flexible retainer packages. Services available as part of a retainer package include market and product analysis and strategy, technology guidance, product evaluation, and merger and acquisition assessment. Even with paid clients, we maintain our strict objectivity and confidentiality requirements. More information on our retainer services (PDF) is available.

- **External speaking and editorial:** Securosis analysts frequently speak at industry events, give online presentations, and write and/or speak for a variety of publications and media.

- **Other expert services:** Securosis analysts are available for other services as well, including Strategic Advisory Days, Strategy Consulting engagements, and Investor Services. These tend to be customized to meet a client's particular requirements.

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

Additionally, Securosis partners with security testing labs to provide unique product evaluations that combine in-depth technical analysis with high-level product, architecture, and market analysis. For more information about Securosis, visit our website: <http://securosis.com/>.