



Endpoint Advanced Protection Buyer's Guide: Prevention PoC Guide

Version 1.5

Released: August 14, 2018

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.

This report is licensed by Carbon Black.

The logo for Carbon Black, featuring the company name in a bold, black, sans-serif font inside a white rounded rectangle.

www.carbonblack.com

Carbon Black (NASDAQ: CBLK) is a leading provider of next-generation endpoint security. Carbon Black serves more than 3,700 customers globally, including 33 of the Fortune 100. As a cybersecurity innovator, Carbon Black has pioneered multiple endpoint security categories, including application control, endpoint detection and response (EDR), and next-generation antivirus (NGAV). Leveraging its big data and analytics cloud platform – the Cb Predictive Security Cloud – Carbon Black solutions enable customers to defend against the most advanced cyber threats, including malware, ransomware, and non-malware attacks. Deployed via the cloud, on premise, or as a managed service, customers use Carbon Black solutions to lock down critical systems, hunt threats, and replace legacy antivirus.

Copyright

This report is licensed under Creative Commons Attribution-Noncommercial-No Derivative Works 3.0.

<http://creativecommons.org/licenses/by-nc-nd/3.0/us/>



Endpoint Advanced Protection Buyer's Guide: Prevention PoC Guide

Table of Contents

Introduction	4
Getting to the Short List	5
What to Test?	7
Rules of Engagement	10
Preparation	12
Phase 1: The Lab Test	15
Phase 2: User Pilot	17
Phase 3: Paid Pilot	19
About the Analyst	20
About Securosis	21

Introduction

Having waded through the extensive selection criteria for endpoint prevention technologies, it is time to see what will work in your environment, which means a Proof of Concept (PoC) test. Shockingly, the reality of how a solution works in your environment may diverge a bit from a demo or even a lab test. So you need to test any solution — especially one costing potentially hundreds of thousands of dollars — in your environment to ensure it will deliver the effectiveness you need, and also that it fits your operational model and meets both security and compliance requirements.

This guide focuses on what you need to know and do to choose the best solution to prevent malware outbreaks in your organization. We'll start with some philosophical perspectives.

1. **You drive the bus:** Many vendors want to bring their testing protocols in and run your PoC. You can do that, but how likely do you think it is that their tool won't shine in their own test? Of course, their tool will shine. It's their test, after all! You want to define the test cases, determine success criteria, and weigh in on the testing protocols. Otherwise, you are outsourcing much of your decision-making to the vendor. This guide prepares you to define the testing process and compare solutions objectively.
2. **One size does not fit all:** Endpoint prevention is one of the security markets with the least variability. With clear similarities between a majority of the malware an organization sees, you could conclude that once you've seen one advanced malware solution, you've seen them all. But you'd be wrong. It comes back to understanding the kinds of malware you'll most likely see and optimizing the testing around that. It's not that other malware isn't important, but focusing the most likely attacks ensures you select the best tool for your environment.
3. **Intangibles matter:** If you won't prevent attacks, why bother? But given minimal differences in effectiveness the intangibles like deployment, endpoint agent overhead, integration with existing security systems, and ongoing management, can make a huge difference. Your testing needs to take these intangibles into account as well.
4. **Fail fast:** We fundamentally believe you cannot learn everything about a tool during a PoC... or any test. So you'll spend a bunch of time figuring out why your results differ when you jump from a dozen to a hundred to a couple of thousand users. You'll benefit greatly from a phased approach, expanding your implementation in measured increments. That way you can get out if something fails — not just your career prospects. You install maybe a hundred, and then perhaps a thousand, to see if you get consistent results. Don't go from a dozen in a test group directly to 100,000 installations. Not if you want to keep your job, anyway.

Getting to the Short List

Before we jump into the specifics of the PoC, revisiting the entirety of the buying process will provide context for where you've been and where you are going. Given a crowded market like Advanced Endpoint Protection abounds with challenges determining the actual capabilities of a product, and how it will meet your requirements.

The following steps should minimize your risk and help you feel confident in your buying decision, although not all these steps are appropriate for every organization. Optimize your buying process based on what helps you make the decision. Period. Here is the full list — pick and choose which make sense for your organization:

Given a crowded market like Advanced Endpoint Protection abounds with challenges determining the actual capabilities of a product, and how it will meet your requirements.

1. **Issue the RFI:** Larger organizations should issue an RFI through established channels and contact a few leading endpoint security vendors directly. If you're a smaller organization, it makes sense to send your RFI to a trusted VAR and email a few endpoint vendors who seem appropriate for your organization.
2. **Perform a paper evaluation:** Before bringing anyone in, match any materials from vendors or other sources against your RFI and draft RFP. Your goal is to build a short list of 3 products which match your needs. Additionally, use outside research sources and product comparisons as appropriate.
3. **Bring in 3 vendors for an on-site presentation and risk assessment:** Nearly every endpoint security vendor will be happy to come in and give you their dog and pony show to inform you of how awesome they are and highlight issues with every other vendor. Use this opportunity to meet directly with the vendors (or your VAR) and get more specific answers to a standard set of questions (you can consult our post for ten questions we like to use).
4. **Finalize your RFP and issue it to your short list of vendors:** At this point, you should thoroughly understand your specific requirements and issue a formal RFP. The formality of the RFP correlates with the size of the deal and the involvement of your procurement folks. Smaller deals may not require a full 40-page set of requirements. Seven-figure deals certainly do. We don't advocate more work than necessary, but for complicated environments, a formal process can minimize challenges to your selection.

5. **Assess RFP responses and begin product testing:** Review the RFP results and drop anyone who doesn't meet any of your hard requirements such as platform support or deployment model (for example cloud-based management). Then bring any remaining products in for in-house testing. The rest of this guide goes into the specifics of the product test.
6. **Select, negotiate, and buy:** Finish testing, take the results to the full selection committee, and begin negotiating with your top choice.

You may wonder how the same products can yield such different results. Some tests show Vendor A stinks. Others show Vendor A in an impressive light. Where is the truth? Probably somewhere in the middle, but ultimately a third-party test has little relevance for you and your selection process.

Before jumping in, a few points about third-party testing data for endpoint security products. If you read a few of the reports, you may wonder how the same products can yield such different results. Some tests show Vendor A stinks. Others show Vendor A in an impressive light. Where is the truth? Probably somewhere in the middle, but ultimately a third-party test has little relevance for you and your selection process.

Best case, you can use third-party tests to whittle down two dozen vendors in the space to five or so to include in your RFI/RFP process. If a solution consistently stinks across all external tests, it probably doesn't provide sufficient effectiveness. But you get the best perspective

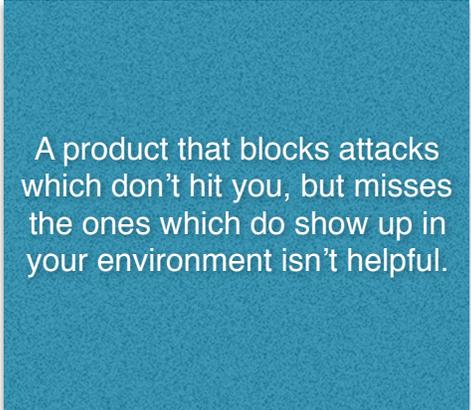
on a product's capabilities by testing it in your environment. That is the point of a PoC.

Once you have 2-3 finalists for your PoC, you should feel confident they all can do the job. The rest of the process focuses on selecting the best product for your organization.

What to Test?

Now that you have finalists you should figure out what you will test. You can start with this list:

1. **Effectiveness:** This is the most visible measure of success for an endpoint prevention solution. It needs to work. But what does that mean, exactly?
 - a. *Adversary context:* A product that blocks attacks which don't hit you, but misses the ones which do show up in your environment isn't helpful. So start with likely attacks you expect to see based on the adversaries you face. If you worry most about ransomware, make sure any product you consider will prevent those attacks. Likewise, if you face nation-state attackers, you may need something a bit more focused on zero-day attacks.
 - b. *Detection:* If a product claims to block certain types of attacks, confirm that it does. Pretty simple, right? You'll want to check its ability to search events and provide a detailed analysis of infection on impacted devices. Make sure to track false positives because both cost time and money. Most of all, ensure you can monitor for false negatives in lab tests — much better to catch too much than to miss a big one.
 - c. *Remediation/Response:* You'll also want to test not just how the product cleans up the messes it finds, but also how it integrates into a broader incident response process. To clean up an endpoint, you might wipe the device as a matter of course. But in any case, you want to ensure the endpoint security product can send information to the network (to possibly quarantine devices), the SIEM (to look for other indications of a similar attack), and to any incident response orchestration and automation offering you have or might get.



A product that blocks attacks which don't hit you, but misses the ones which do show up in your environment isn't helpful.

d. *Additional capabilities:* Advanced endpoint prevention products may include additional controls beyond malware detection, including application control (whitelisting), device control, file activity monitoring, etc. These add-on functions can facilitate prevention, such as by blocking execution of unauthorized software (application whitelisting) to limit the ability of malware to compromise devices. So it makes sense to test these add-ons as well.

2. **Deployment:** You already have something running on every endpoint, so pay attention to how easily the product deploys, especially at organizations with many thousands of devices.

a. *Installation:* This includes not just installing the new agent, but also uninstalling the old one. How clean is the transition? How long does it take? Can you minimize disruption? Again, with thousands of devices, every few minutes represents significant opportunity cost savings.

b. *Setting policies:* You already have policies in your existing product for alerting, remediation, user groups, detection sensitivity, etc. Making sure you can quickly get those policies working with the new product will save time.

c. *Integration:* You probably have other security tools running in your environment. So how does the product integrate with your directories (which are important to your policies), your SIEM, Ops' work management/ticketing system, and your network security platform? Optimally you'd like some formal partnership and certification for the integration, and lacking that you need to understand how much work you'll have to do building and maintaining integrations.

d. *Mass deployment:* You can deploy the agent on a dozen machines pretty quickly, even leveraging Sneakernet. But what about 100,000? You'll want to see and experience the deployment tools — your project depends on your ability to install the product on all devices with minimal issues.

You may remember the olden days when security agents consumed more device resources than business applications. That didn't work very well.

3. **Impact on Devices:** You may remember the olden days when security agents consumed more device resources than business applications. That didn't work very well, so you'll want to ensure the product minimizes overhead and drag on devices. You will quantify some of the metrics in the lab and early pilot deployments. You'll also want to perform some qualitative interviews with test subjects to get their sense of device drag, especially compared to the existing agent.

4. **Scalability:** You test products first in a lab, and then with a small pilot deployment. The difficulty arises when determining whether a product's performance with a hundred users matches performance with 10,000. You'll need an understanding of the product architecture, and you may also need to perform background checks with other customers.
5. **Ongoing Management:** Given the dynamism of the threat landscape, you will spend a lot of time in the management interface of your chosen tool. Triaging alerts, remediating, adapting policies, deploying new agents, and about a hundred other tasks will happen in the management console. Do you like the user experience? Can you customize it to make it work for your needs? Does it provide all the capabilities you need, at the scale you need? Once you pick a tool you will live with it, so make sure your PoC puts it through its paces.

Rules of Engagement

Before you start the PoC, you need to define its rules of engagement — at least with your vendors. Also, consider other internal and external influencers who need to buy into your approach. These may include any reseller involved in the deal, as well as internal constituencies such as security operations, IT operations, and senior management to approve the participation of pilot employees.

Defining and agreeing to rules of engagement before the PoC starts means managing expectations with key influencers for the project. We harp on the importance of managing expectations during every step of the PoC (and pretty much everything security) because the easiest way to get in trouble remains not meeting the (all-too-often unspoken) expectations of the parties who work with you.

1. **Define Success:** The vendor, of course, needs to understand what success means, and so does your internal team. So the first big question to agree on is the definition of a successful test. Is it binary, — the product either blocks attacks or not? Is there a grading scale? If so, what is it? Being squishy about success helps nobody — you will need to justify your recommendation, so lean towards quantifying everything. The more structured you make your evaluation criteria, the less heartburn you'll have later in the process. Trust us on this.
2. **Test Phases:** We advocate a 3-stage test for a security tool used by pretty much every employee. Yes, it takes longer. Yes, it's a pain in the backside. But you need to both make sure the product works for your entire organization, and pre-sell it to the folks who need to use it. Don't underestimate the importance of getting buy-in from critical internal groups. They can screw up your procurement — trust us on this too.
 - a. *Phase 1: Lab Test.* First, you'll install the product in the lab to get a sense of effectiveness, deployment, and management by playing around with it. You can also use nasty malware in the lab which you would never play with on a production network. You need to decide whether you will test multiple products in the lab. You likely will, but it's worth considering how many resources you can devote to your top choices.

The first big question to agree on is the definition of a successful test. Is it binary, — the product either blocks attacks or not? Is there a grading scale? If so, what is it? Being squishy about success helps nobody.

- b. *Phase 2: User pilot.* After the lab test the survivors enter a stage of actual deployment with real employees. You'll want to test the deployment and performance drag on devices in use by actual people, as well as make sure the product blocks real attacks after those real people click malicious links. You can't do that in a lab. Again, you'll need to figure out whether you want to introduce multiple tools into a user pilot. It should be obvious, but don't install multiple products on the same user systems. Figuring out which tool detected which attack would hamper the testing process.
 - c. *Phase 3: Paid, broader pilot.* Most vendors want you to sign on the proverbial dotted line after the user pilot. We suggest performing a more extensive pilot before committing to deploying across the enterprise. If you can get the vendor to expand the deployment to a couple of hundred users as part of the PoC, you've done an excellent job! But expect to pay for those seats, because you'll want to experience the product as a real live customer.
3. **Timeframes:** You also need to be clear on how long you expect each phase to take. Depending on the sophistication of your lab (assuming you have one), you can expect to spend 1–2 weeks putting a product through its paces in the lab, and possibly more, depending on how much malware you want to throw at it. You'll want to set aside at least 30 days for the user pilot because it involves actual employees being attacked and clicking bad stuff. Your adversaries probably don't work on a strict schedule, so you can't guarantee they'll attack during an arbitrary testing window. The broader pilot should be long enough for you to get a sense of whether the tool works for your environment. You'll need to strike a balance between doing enough work to make sure the tool does the job, and the fact that every day you don't go to full deployment, you probably depend on old technology to protect your enterprise.

At this point, you know what to test, and you have built a consensus on the structure of your PoC. The time has come to get to work. The next phase of the PoC involves preparing for the test.

Preparation

You know the old saying: “If you fail to prepare, you prepare to fail.” As you roll your eyes at the security curmudgeons who have seen it all (yes, that’s us), you know it’s true. This section offers some detail about preparing for the PoC.

Building the Testbed/Lab

As mentioned above, the lab test kicks off the first phase of the PoC. For a lab test, you need a lab. So you need to build a lab. First, you need to figure out what kind of malware you want to test. Given the infinite number of samples, let’s group them:

1. **Standard malware:** These are attacks you’ve seen before. But many products still use signatures in some form, so you’ll need to change the hash of the malware (mutate it), which requires a packer of some sort.
2. **Disconnected Devices:** Not every device can be updated in real time, so test whether devices catch malware even if the product hasn’t updated for a couple of days. Using a device that missing recent updates can also simulate an unknown attack. We don’t consider this a zero-day — actually to test a zero-day you need to have one. And unless you work for the NSA, you probably don’t. In effect, try last week’s product to stop today’s malware. That will show you pretty quickly whether the product can handle attacks it hasn’t seen before.
3. **File-less malware:** As we discussed in this post, attackers aren’t using files exclusively anymore. They can hide attacks in the registry or authorized applications like PowerShell. Many sites aggregating malware for testing also provide malicious scripts to run on victim device as part of the test.

So where do you get the samples? That’s a bit of a moving target, but if you have a forensic team in-house, they should have some good stuff. There are also both open source and subscription sites which can provide malware files and file-less attack samples.

Vendors can also provide malware samples to test. They use packers (as described above) to change the hashes of samples, so they are “like new.” Except they aren’t. A bit of an endpoint security scandal erupted a while back when a vendor allegedly put markers in malware samples they delivered for customer tests which identified their samples as malware, regardless of hash. Amazingly, that vendor aced all tests against malware they provided. If you want to get malware samples from vendors, at least use one vendor’s samples against another vendor’s product. But sourcing your own malware samples remains a more reliable option.

Once you have a plan to gather malware for tests, you need somewhere to run them, and that's the lab.

1. **Victim Devices:** You need devices on hand to run tests. That means a variety of devices seen in your environment, including multiple versions of Windows and probably some Macs. Malware can also target servers, so you may also want some Windows Servers ready to test, as well as Linux if you plan to cover those with a prevention product.
2. **Attack Devices:** These devices launch attacks when testing against a 'live' adversary. They run attacker tools (for a list, consult any red team penetration testing guide) and launch an attack against a known vulnerability.
3. **File Server:** Malware (especially ransomware) targets file shares, so you need a file share available on your lab network to test the compromised device's attempt to perform reconnaissance, identify a file share, and encrypt files.
4. **Network:** Many of these products work do not require an on-premise server. They connect to the "mothership" (a cloud-based service) for initial deployment, product updates, and policy changes. You need to be able to turn the network on and off to simulate remote offline use — not every employee connects to the network at all times, but they still need protection.

Don't exclusively using virtual machines in the lab. Some advanced malware does not execute on virtual devices. You'll want a mix of virtual and physical machines for a complete test.

We like using VMs in the lab because they provide flexibility for running operating system versions you may not otherwise have running or accessible, as well as leverage to test more devices without extensive hardware. But don't exclusively using virtual machines in the lab. Some advanced malware does not execute on virtual devices. You'll want a mix of virtual and physical machines for a complete test.

You may consider using a testbed in the cloud provided by a vendor. Using a vendor environment can save you from having to build a lab yourself, but you cannot get

consistent results between vendors if you use their respective cloud testbeds, because you would be running tests in different testbeds.

Instrumentation and Data Gathering

All these tests don't help much if you don't collect sufficiently granular data on the success of the attack or the effectiveness of remediation. You need to instrument the testing devices to gather data on the actions malware takes on each device, any changes to device configuration or registry, and network captures from each device. Fortunately, endpoints generate a ton of log data about all sorts of device events. Capture those logs on the test machines as a start. For more detailed telemetry,

including memory maps, you'll want to look at open source forensics tools which track endpoint activity and can detect successful compromise.

You can also use Endpoint Detection and Response (EDR) technology on testing endpoints to gather very detailed telemetry. A novel use of EDR, right? As with vendor malware samples, you may want to use one vendor's EDR package to test another vendor's prevention. We're a bit paranoid, but over the years we've seen enough PoC shenanigans to be cautious.

You also need to track the performance impact of these prevention products on your devices. Traditional endpoint management tools offer very granular device usage/activity information. To examine resource usage you'll want a baseline without any prevention technology, and another using your existing product. With these, you can make fair comparisons for each product's overhead.

Choosing Candidates for the User Pilot

We don't want to put the cart before the horse, but you also need to determine which users will participate in the pilot.

1. **Diversity:** For the user pilot we recommend a reasonably diverse user community. Many organizations default to IT folks, who like to play with new toys and can be forced to participate (we're kidding, kind of). We don't have a problem with using a few IT folks, but also include less sophisticated employees and different business units. Attackers target Finance, HR, and R&D, and those employees use different applications, so they should be included in the test. Don't forget remote users. It's a hassle to include them, but unless all employees work in the office, you need to know how the product holds up for remote users.
2. **Number of Participants:** You want enough users in the pilot to get real data, but not so many that it becomes unwieldy. For most organizations, we see user pilots between 25 and 100. You want the minimum number of users to provide a diverse cross-section of employees. More than that complicates your PoC unnecessarily.
3. **Infection Baselines:** Before you include a user in the pilot, make sure you have data about the number of infections for that specific user over the past 30, 60, and 90 days. You need to know (and prove) whether the tested product blocks more attacks, which requires you to know how many attacks the user(s) saw over the relevant time periods.

Phase 1: The Lab Test

It's showtime! Once you have a lab set up, you can start testing. As mentioned above, having precise and measurable (where possible) criteria makes the PoC run a lot more smoothly.

Initial Install

Here focus on whether the tool removes the existing agent and then installs cleanly and completely, how well it fits into your current software distribution and update processes, and ultimately how much work it takes to make the switch. You'll test conflicts and challenges on employee devices during the user pilot, so focus here on whether the product can install on the devices in your environment.

Lab Testing Effectiveness

Determining whether the product blocks attacks demands most of your effort during the lab test. There are a variety of testing protocols for running a product through effectiveness tests, so we won't go into detail here. Prioritize the following areas:

1. **Files:** Look for malware loading into the file system of the device. Does detection/blocking happen before writing the file to disk? What happens when files change? You'll want to use common malware and mutated versions of files to get a sense of whether the product has limitations when signatures don't match exactly.
2. **Running attacks:** You'll also want to run attacks against devices using a variety of exploits to see whether you can compromise a machine using known exploits. These should include publicly available exploits, including file-less and registry attacks. For consistent results across products, script your attacks.
3. **Clean-up/remediation:** Understanding products sometimes miss, also test how well the product cleans up a device after compromise. Run malware against a clean and unprotected machine and then install the agent to see what it finds and whether it can clean it up.

We need to address turning off protections during testing to assess a product's strengths in static file analysis versus dynamic detection. Some third-party testing groups do this to rank different products. This provides limited relevance to enterprise lab tests. If you plan to shut off pre-execution protection on employee devices in practice, or for that matter, something like device control, then feel free to shut it down during the test. But we have wonder why you would do that. For our recommended PoC approach, test the product as it will run in your environment, with all its capabilities.

If you plan to shut off pre-execution protection on employee devices in practice, or for that matter, something like device control, then feel free to shut it down during the test. But we have wonder why you would do that.

Performance Impact

In the lab, you'll benchmark device performance impact as well. You can do this by baselining device performance without any protection, with the existing agent (for comparison), and with the new product installed. You can use any of the publicly available device benchmarking tools (Google "benchmark your PC") and get measurements for a qualitative sense of the agent's device impact.

You can also benchmark network traffic by capturing the packets from the device over an arbitrary amount of time and comparing with traffic from the existing agent.

Phase 2: User Pilot

Once you finish Phase 1, test the product in your real environment. That means rolling it out to a group of employees. Make sure you have data about the infections and issues each tester experienced over the past couple of months. Otherwise, you will have no idea whether the new product performs better or worse.

Deployment

For the user pilot, you need a sense of how quickly and efficiently the product will go into your environment. That involves integrating the tool with your directory and setting up users and policies. Then go through deployment/install, paying attention specifically to conflicts and broken applications. The test subjects will let you know when something breaks, which impacts their ability to work.

Don't forget remote users. Did the product install when they weren't on the corporate network? Were there any issues the help desk had to handle? Given the increasing percentage of remote/disconnected users, the product needs to support folks outside of the corporate office completely.

We discussed instrumentation above, and it comes into play again here. You'll want to gather as much device telemetry as you can.

Soaking

After installing the product on the employee devices, you wait. Statistically (if you picked the user pilot group correctly), a number of those devices will suffer attacks, and the product will work. Or it won't. You get detailed telemetry off each device, so you'll know what happened and whether prevention worked.

While the product soaks in your environment, you can test out the intangibles we described above. Change the policies and see how that goes. Does it impact users at all? Test the integrations with your security operations tools. Run the compliance reports past your assessor to make sure they get the substantiation they need. Assuming the products test out similarly in effectiveness (maybe a crappy assumption, but go with it for the moment), things like reporting and ongoing management will determine the ultimate winner.

These are production devices, so you may want to monitor the network more closely. If you miss an attack, or the product is deficient in some other way, you don't need a real incident to prove a product doesn't work. An actual live infection does provide an opportunity to test remediation capabilities. Gather qualitative information about handling the compromise. Did it work? Was the remediation experience as you expected?

Qualitative Assessment

The final set of questions to answer evaluates employee experience with the product. What did they notice about the new tool? Was it noisy? Did they see when the product blocked attacks? Did the product drag performance of their device down at all? Did any applications or other functionality stop working after installing the new agent? Employees can fill out a survey and provide an assessment for each question.

Remember that a lot of these qualitative questions are subjective, so treat the results accordingly. Focus on trends. Do all users complain about performance? Just in some groups? Are there business-critical applications the product flags as malware? That's been known to happen, and negatively impact the perception of a tool.

This sniff test will identify if the users have uncovered a deal-breaker that would impact deployment across the broader user community.

The final set of questions to answer evaluates employee experience with the product. What did they notice about the new tool? Was it noisy? Did the product drag performance of their device down at all? Did any applications stop working after installing the new agent?

Analyzing Results

Once you have all the quantitative data on how the product performed with test subjects you can make comparisons. Did you have demonstrably fewer infections? Were they cleaned up effectively? Go back to the success criteria you defined at the beginning of this process to ensure you can communicate the effectiveness of the product you tested.

Next Step

After repeating the testing process for the other products under consideration, now you need to choose. If you have confidence, the product will catch more and disrupt less, and your existing product doesn't catch much at all, then you can select the winner and skip right to full deployment. After a short and easy negotiation with the vendor, right? OK, maybe not so easy, but hopefully short.

But if you have any misgivings about the products, we first recommend a broader deployment focused on scale and operational fit within your organization. Understand this additional test will cost money. You need to license the small deployment and will need to provide similar instrumentation to the user pilot, which may involve additional licensing fees.

A typical enterprise might spend high six figures or more on a full deployment. But you can spend some money on a limited implementation, covering perhaps 10% of your user base. You need to phase in the production deployment anyway. It's not like you would install 100,000 systems with the new agent over a weekend, so consider the paid pilot the first wave of the production roll-out, enabling you to refine deployment.

Phase 3: Paid Pilot

As you go from perhaps 100 users to the first phase of production deployment, pay attention to consistency. Has effectiveness been consistent? Has the new agent broken any other applications? Do you have sufficient policy control and flexibility to meet the needs of the expanded user base?

Have alerts become overwhelming? Did the product miss (false negative) at the same rate as the user pilot?

Phase 3 not just proves the concept, it validates the decision and the roll-out. Assuming all goes well, you have justified the vendor choice and proven the scale of the solution (to the degree that's possible).

Pay particular attention to product updates and network traffic needs. Starting with maybe 100 users for the initial pilot, consumption of network capacity was likely minimal. Is that still the case? And what about necessary product updates? Does that bog down the network or devices?

Phase 3 not just proves the concept, it validates the decision and the roll-out. Assuming all goes well, you have justified the vendor choice and proven the scale of

the solution (to the degree that's possible). Then it's just about finalizing procurement and planning the roll-out to the entire organization.

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

About the Analyst

Mike Rothman, Analyst and President

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

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

Mike published [The Pragmatic CSO](http://www.pragmaticcco.com/) <<http://www.pragmaticcco.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](mailto:mrothman@securosis.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:

- **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 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/>>.