Endpoint Advanced Protection Buyer's Guide: Detection and Response PoC Guide

Version 1.3

Released: August 27, 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.

Carbon Black.

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

Endpoint Advanced Protection Buyer’s Guide: Detection and Response PoC Guide

Table of Contents

Introduction

Having waded through the extensive selection criteria for endpoint detection and response technologies, now you need 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. Thus 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 on the PoC process.

1. **You drive the bus:** Many vendors want to bring their testing plans and processes to the table and run your PoC. You can do that, but how likely do you think it is that their tool won't shine in a test they create? 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 and giving them a very favorable situation to ace the test. This guide prepares you to define the testing process and compare solutions objectively.
2. **One size does not fit all:** Endpoint detection and response tools provide capabilities to undertake a complicated function within organizations. You have to consider not just the technical capabilities of the provider, but also the skills of your staff. Having an electron microscope to study an infection will provide significant visibility into the sample, but not if your folks don't have the capabilities to utilize the enhanced visibility. It comes back to understanding how you want the detection and response process to run in your environment and optimizing the testing around that.
3. **Intangibles matter:** You need a tool that provides the capabilities required, but don't forget to consider intangibles like deployment, endpoint agent overhead, integration with existing security systems, and ongoing management that can make a huge difference.
4. **Fail fast:** We fundamentally believe you cannot learn everything about a tool during a PoC or any test. We think you'll benefit significantly from a phased approach to deployment, expanding your implementation in measured increments. That way you can get out if something fails. You install maybe a hundred, and then perhaps a thousand, to see if the detection and response process scales. We caution you from going from a dozen in a test group directly to 100,000 agents in use. 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. A crowded market like Advanced Endpoint Protection abounds with challenges determining the actual capabilities of a product, and how it will meet your requirements.

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 the 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:

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 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 the selection criteria guide 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 very likely do. We don't advocate more work than necessary, but for complicated environments,

a formal process can minimize challenges to your selection. Especially those challenges that happen after the fact.

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.

> Regarding response and hunting the comfort of the team using the tool should sway the decision because making the team more effective and efficient provides much of the value of EDR.

Before jumping in, a few points about third-party reviews for endpoint detection and response products. Amongst security products, evaluating EDR remains very subjective. Yes, you want top tier detection. But that's only the start of the response process. Regarding response and hunting the comfort of the team using the tool should sway the decision because making the team more effective and efficient provides much of the value of EDR.

Third party reviews can surface the strengths and weaknesses of each tool, and we have no issue with that. Just keep in mind the subjectivity of a reviewer's perspective on validating an alert, doing malware analysis, or something very individualized like hunting. You get the best perspective on a product's capabilities by testing it in your environment. That is the point of a PoC. 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.

Once you have 2-3 finalists for your PoC, you should feel confident that any of the finalists 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 you'll need to design the test. This list mostly draws on the research in the Detection and Response Selection Criteria Guide. You can refer to that document for more detail on the specific capabilities.

1. **Detection**

   a. *Effectiveness:* The most visible measure of capability for detection tools, you'll need to confirm the product detects the types of attacks you face. 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.

   b. *Policy Granularity:* When it comes to detection, significant variances exist between the risk presented to your organization by different classes of devices. Thus, you'll want to make sure you can quickly set different types of policies for different categories of devices.

   c. *Threat Intelligence:* Detecting modern attacks requires staying current regarding new attack techniques. Built-in security analytics provides some of the capabilities to detect unknown attacks, but it's also useful to integrate 3rd party data into the system, preferably in an automated fashion. External security data sources may identify attacks that you don't see, and to learn from the experiences of other organizations.

   d. *Risk Scoring:* With dozens of alerts hitting daily — perhaps significantly more — it is essential to weigh which alerts warrant immediate investigation, and a risk score should inform your decisions. Focus in the PoC on your ability to tune the scoring environment to your environment.

Attackers typically orchestrate multi-faceted attacks involving multiple tactics across many devices to achieve their missions.

   e. *Campaigns:* Attackers typically orchestrate multi-faceted attacks involving multiple tactics across many devices to achieve their missions. To detect these more sophisticated attacks requires aggregating telemetry across devices and looking for signs of a coordinated attack (or *campaign*). You'll need to test the ability of the product to visualize the activity across the entire environment and to drill down into specific devices to validate the attack.

    f. *Enriching Alerts:* Adding adversary information and assembling available artifacts can accelerate the response process. When the analyst (typically in the response process) opens up an enriched alert for validation, most (if not all) of the information they'll need is there at their fingertips.

    g. *Cloud:* Given the increasing cloud centricity of EDR tools, you'll want to test storage usage, network impact of moving large amounts of telemetry to the cloud, and the ability to set policies and manage the environment via the cloud-based interface.

2. **Response and Hunting:** Significant overlap exists between response and hunting tools, so we'll cover both in this section although you'll test each function at different points in the PoC. The main difference between response and hunting has to do with what triggers action. For the response process, alerts act as the catalyst for action. For hunters, they've come up with a hypothesis for how the attackers compromised a device, and they need to prove (or disprove) it using telemetry gathered from the endpoints.

    a. *Analyst experience:* Individual responders have their own way of doing things, and you want the tool to accelerate and facilitate their process — not break it. Thus, usability should be the primary focus of the PoC. Having a sophisticated endpoint response platform doesn't help if the analysts don't use it.

    b. *Chain of custody:* In some cases, security data from the attack may end up as evidence in a court proceeding. That means you need to pay attention to the chain of custody and the device imaging process, etc.

Individual responders have their own way of doing things, and you want the tool to accelerate and facilitate their process — not break it. Thus, usability should be the primary focus of the PoC.

    c. *Alert Validation/Triage:* Make sure that the analyst(s) have all of the information they'll need to validate the alert within the case from the start. The less time the analyst needs to gather related data, the less time they spend identifying root cause and responding to the incident. This function involves sufficient alert enrichment, case management, visualization, drill down/pivot, and workflows/automation to keep the analyst productive.

    d. *Containment/Remediation:* You'll also want to test integrations with enterprise controls, including network security so that the analyst can quarantine a compromised device directly from the response console. You'll also need to confirm the ability for the tool to kick off a remediation process, either by providing direct access to the device or integrating with an endpoint management tool.

e. *Search:* During the response process your team will rely heavily on search, looking for specific indicators of compromise on devices in your environment. Test that you can mine significant amounts of endpoint security data without knocking down the system or grinding performance to a halt.

f. *Retrospective search:* A capability very useful to response, retrospective search allows responders to search through historical telemetry from the organization's endpoints enabling them to find malicious activity which might not have triggered an alert at the time, possibly because it wasn't then a known attack. You use this capability during a response to gauge the proliferation of an attack.

3. **Deployment:** You already have a endpoint security agent running on every endpoint, so pay attention to how quickly and easily the product deploys, especially at organizations with many thousands of devices.

   a. *Installation:* This includes installing the new agent and making sure no conflicts emerge with any other endpoint security agents (or other software) on the device.

   b. *Setting policies:* You already have policies in any existing endpoint security products for alerting, remediation, user groups, detection sensitivity, etc. Make sure you can quickly get those policies working with the new product.

   c. *Integration:* You very likely have other security tools running in your environment. How does the product integrate with your directories (which are essential to your policies), your SIEM, the operations team's work management/ticketing system, and your network security platform? Optimally you'd like some formal partnership and certification for the integration. 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.

4. **Impact on Devices and Infrastructure:** You'll want to ensure the product minimizes overhead and drag on devices. Endpoints also create a significant amount of telemetry that needs to move from the device to an aggregation point for analysis. 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 any existing agent running on their devices.

5. **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, especially searching for indicators. You'll need an understanding of the product architecture, and you may also need to perform background checks with other customers.
6. **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. Does the team 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 for a while, 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 and any 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 the 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. The first big question to agree on is the definition of a successful test. Given the subjectivity inherent to response and hunting, being squishy about success helps nobody. Don't forget you will need to justify your recommendation at some point, 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.

   Success within the context of testing effectiveness is very similar to testing a prevention tool. Does it find the attack you are likely to see? When testing response and hunting, you have two objectives. First, you need to ensure the tool works for your process and helps the team. If it doesn't help, the tool (and the test) fails.

   Additionally, you can get educational value by having some less experienced responders and hunters participate in the PoC. This way the junior folks get exposure not just to the tools, but also to the process of responding and hunting. Hands-on responding (or hunting) can help them improve their skills and justify testing out a few EDR tools.

When testing response and hunting, you have two objectives. First, you need to ensure the tool works for your process and helps the team. If it doesn't help, the tool (and the test) fails.

2. **Test Phases:** We advocate a 2-stage test for an EDR tool used typically by folks in the SOC.

    a. *Phase 1: Lab Test.* First, you'll install the product in the lab to get a sense of deployment, management, and user experience by playing around with it. You can also ensure adequate detection capabilities by using nasty malware in the lab which you would never play with on a production network.

    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 detects real attacks after those real people do bad things (or have bad things done to them). You can't do that in a lab. You can also do a live hunting test after deploying the agents on the test machines. Not much beats (in security anyway) finding heretofore unknown adversaries in your environment during a PoC.

        Again, you'll need to figure out whether you want to test multiple tools in a user pilot. It should be obvious, but don't install multiple products on the same user systems. Trying to figure out which product detected which attack would hamper the testing process.

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 and your response to those attacks. Your adversaries probably don't work on a strict schedule, so you can't guarantee they'll attack during an arbitrary testing window. 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, and you’ll be testing whether the product detects attacks. Thus, you’ll need to build a lab if you don’t already have one. First, you need to figure out what kind of malware you want to see if the tool detects. Given the infinite number of samples, let’s group them:

> You’ll need to build a lab if you don’t already have one. First, you need to figure out what kind of malware you want to see if the tool detects.

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 detect malware even if the product hasn't updated for a couple of days. Using a device 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 the product without an update and see whether it can detect the most recent malware.

3. **File-less malware:** Attackers aren’t using files to spread malware exclusively anymore. They can hide attacks in the device’s registry or by using authorized applications like PowerShell. Many sites aggregating malware for testing also provide malicious scripts to run on victim devices 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 malware samples yourself 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.

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.

Testing Campaigns and Response

As you recall, a campaign involves the adversary compromising multiple devices to achieve their mission. The best way to test these capabilities is to stage a red team effort where you set up a target and have the penetration testers go after it. You'll need to set up a target file somewhere on the lab network, and a variety of devices on logical network segments to mimic a small network.

The best way to test campaign and response capabilities is to stage a red team effort where you set up a target and have the penetration testers go after it.

Staging a number of machines to participate in a campaign scenario does add a lot of complexity to the test network, yet we don't see another option to detecting complicated attacks and evaluating how your team responds to it.

Going on the Hunt

You don't have to do a lot of prep for a hunting PoC because basically, you let the hunters hunt within your environment. The process of hunting doesn't make changes to the devices, instead looks for indications of malicious activity that would then trigger a response process if found. We don't see a lot of downside in hunting on your networks using tools you basically "borrow" during the PoC. What's the worst that can happen? You find an active adversary that you didn't know about? That's a pretty good outcome in our book.

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 and the information provided by the EDR tool to facilitate response. 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 also need to track the performance impact of these detection and response 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 EDR agent installed and another using the EDR product. With this data, 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 involved 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, especially relative to accessing telemetry on their systems.

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 can detect new attacks better than an old tool, 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 communicating the results of the PoC a lot easier.

Initial Install

Upon installing the product ensure it doesn't step on any other agents (typically EPP or Advanced Prevention) already on the device. You'll also want to test how well it fits into your current software distribution and update processes, and ultimately how much work it takes to deploy the tool. You'll do a broader test of software conflicts on employee devices during the user pilot, so focus here on whether the product can install on the majority of devices in your environment.

Upon installing the product ensure it doesn't step on any other agents (typically EPP or Advanced Prevention) already on the device.

Lab Testing Detection

Determining whether the product detects attacks demands most of your effort during the lab test. There are a variety of protocols for running endpoint security products through detection effectiveness tests; we won't go into detail here. Prioritize the following areas:

1. **Files:** Look for malware loading into the file system of the device. When does detection happen? Before the malware writes anything to disk? Earlier? What happens when system files change? You'll want to use common malware and mutated versions of files to get a sense of whether the product can detect malware packing and other means of obfuscating malicious activity.

2. **Running attacks:** You'll also want to run attacks against devices using a variety of exploits to see whether the tool will detection a compromise in progress using known exploits. These should include publicly available exploits, including file-less and registry attacks. For consistent results across products, script your attacks.

Lab Testing Response

As we described above, the best way to test the response capabilities of any product involves *responding* to an incident. You'll need to set up a device and have your red team attack it on the lab network. This way you'll see what's happening on the device, what the EDR tool is detecting, and allow your folks to run the response process with the new tool.

Ensuring the tool works for your people and process is the most important output of the PoC. Not that your folks should adapt their process where it stands to improve, but you want to the tool to make the response process more effective and efficient.

Remember that ensuring the tool works for your people and process is the most important output of the PoC. Not that your folks should adapt their process where it stands to improve, but you want to the tool to make the response process more effective and efficient.

Pay attention to how quickly your team can validate the attack, identify the root cause, determine the extent of the compromise. On a small lab network, you won't be able to truly understand how the tool works at scale (you'll do that during the user test), but your people can assess qualitatively whether the tool helps or doesn't.

Performance Impact

In the lab, you'll benchmark device performance impact of the EDR tool 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, you'll then 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 before you start the user test. Otherwise, you will have no idea whether the new product detects the attack better or worse than the status quo.

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? Given the increasing percentage of remote/disconnected users, the product needs to support folks outside of the corporate office completely.

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.

Waiting for the Attack

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 detect the malicious activity. Or it won't. You get detailed telemetry off each device, so you'll know what happened and whether prevention worked.

Upon detecting an attack, you can launch your response process using the EDR tool, as an adjunct to your other response tools.

While the product soaks in your environment, you can test out the intangibles we described above. Change the detection 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 detection effectiveness, things like reporting and ongoing management will determine the ultimate winner.

Testing a “Real” Response

You put the product through its response paces in the lab test (thanks Red Team). You could do another test utilizing the red team, basically setting them loose on the actual user devices. Although senior management may frown upon intentionally disrupting the work of an employee doing their job.

So basically you wait for an attack and then respond accordingly. You’ll want to gather qualitative information about handling the attack. Was the response experience as you expected? As we proposed above, ask the responders whether the tool helped or hindered their response.

If the answer was not a resounding, “What an awesome tool,” you probably should scrutinize the purchase a bit more. You already should have some prevention capability on the device, so unless EDR demonstrably adds value to your team, it may not warrant an additional investment.

Staging a Hunt

You may not have done a real hunt during the lab test, mostly because your lab resources were better spent testing detection and response. But now that you have agents deployed on the devices, your team can find some active adversaries. We’re not going to go through the mechanics of hunting in this Guide, but you’ll want the team to start with a hypothesis to look for specific attacks.

The effectiveness of a hunting tool lies in making the hunters more efficient and streamlining their efforts. Evaluating a hunting tool can be more subjective, and that’s OK. It’s a bit harder to compare apples to apples within the context of a hunting PoC.

Then the hunter exercises the search capabilities of the tool extensively to see if any of those particular indicators show up. If you do get a hit, then you use the EDR tool for validation and triage of the potential attack. Finally, if it turns out to be an active adversary, you’ve already gathered a lot of information for the case and can move directly into a response process.

As we’ve mentioned before, similarities abound in hunting and response tools. The main difference is that an alert triggers the response process while a hunt is more proactive, where the hunter starts with an idea of what he/she wants to test and then goes looking for that activity.

The effectiveness of a hunting tool lies in making the hunters more efficient and streamlining their efforts. Evaluating a hunting tool can be more subjective, and that’s OK. It’s a bit harder to compare apples to apples within the context of a hunting PoC.

Qualitative Assessment

The final set of questions to answer evaluates employee experience with the product. What did they notice about the new tool? 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 of 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 impacts? 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.

Analyzing Results

Once you have all the quantitative data on how the product performed with test subjects you can make comparisons. Did you detect the attacks you know to have happened during the test? If there were attacks, did the response go as planned? Could you find root cause quickly and efficiently, relative to the existing process? Did your hunting test find anything?

Most importantly, do your analysts and other security team members like the tool? Can they say specifically (and backed by data) that the tool will help them do their jobs better? 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.

Wrapping Up the PoC

After repeating the testing process for the other products under consideration, now you need to choose. After what probably has been an exhaustive test, the leading product should detect the attacks happening in your environment and provide minimal disruption. The EDR PoC process has also given you an opportunity to test the response and hunting capabilities of the tool and hopefully made your team more efficient and effective.

If you have any misgivings about the products, we first recommend a broader deployment focused on scale and operational fit within your organization. If your user test was 20 employees, maybe you deploy to 100 or 200 higher profile devices and let the product soak a bit more in your environment. Understand this additional test will cost money. But spending a little more and testing a little longer can provide significant dividends if you don't feel good about the purchase. Remember, you put your head on the block when you push for an expensive tool. You want a clear success.

> If you have any misgivings about the products, we recommend a broader deployment focused on scale and operational fit within your organization. Maybe you deploy to 100 or 200 higher profile devices and let the product soak a bit more in your environment.

To be clear, getting to a clear technical winner does not necessarily mean that's the tool you will buy. Negotiation and ultimately procurement may disqualify your leading candidate due to many potential reasons, most out of your control. Which results in our recommendation to evaluate multiple tools and get comfortable that more than one product can meet your requirements. It's always good to have Plan B when buying tech products, especially endpoint security products.

Good luck with your PoC and purchase. 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.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:

- **Primary research publishing**: We publish the vast majority of our research for free through our blog, and package the research as papers that can be licensed for distribution on an annual basis. All published materials and presentations meet our strict objectivity requirements, and follow our Totally Transparent Research policy.

- **Cloud Security Project Accelerators**: Securosis Project Accelerators (SPA) are packaged consulting offerings to bring our applied research and battle-tested field experiences to your cloud deployments. These in-depth programs combine assessment, tailored workshops, and ongoing support to ensure you can secure your cloud projects better and faster. They are designed to cut months or years off your projects while integrating leading-edge cloud security practices into your existing operations.

- **Cloud Security Training:** We are the team that built the Cloud Security Alliance CCSK training class and our own Advanced Cloud Security and Applied SecDevOps program. Attend one of our public classes or bring us in for a private, customized experience.

- **Advisory services for vendors**: We offer a number of advisory services to help our vendor clients bring the right product/service to market in the right way to hit on critical market requirements. Securosis is known for telling our clients what they NEED to hear, not what they want to hear. Clients typically start with a strategy day engagement, and then can engage with us on a retainer basis for ongoing support. Services available as part of our advisory services include market and product analysis and strategy, technology roadmap guidance, competitive strategies, etc. Though keep in mind, we maintain our strict objectivity and confidentiality requirements on all engagements.

- **Custom Research, Speaking and Advisory**: Need a custom research report on a new technology or security issue? A highly-rated speaker for an internal or public security event? An outside expert for a merger or acquisition due diligence? An expert to evaluate your security strategy, identify gaps, and build a roadmap forward? These defined projects bridge the gap when you need more than a strategy day but less than a long-term consulting engagement.

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. For more information about Securosis, visit our website: <http://securosis.com/>.