The Future of Security Operations

Version 1.4
Released: February 21, 2018

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.

Copyright

The Future of Security Operations

Table of Contents

Behind the 8 Ball

The velocity of technology infrastructure change continues to accelerate, putting serious stress on Security Operations (SecOps). This has forced security folks to face the fact that *operations* has never really been their forte. That's a bit harsh, but denial never helps address problems. The evidence is fairly strong that most organizations are pretty bad at security operations. How many high-profile breaches could have been avoided if one of many alerts was acted upon? How many attacks were made possible by not having properly patched servers or infrastructure? How many successful compromises resulted from human error?

Security folks have to face the fact that operations has never really been their forte. That's a bit harsh, but denial never helps address problems. The evidence is fairly strong that most organizations are pretty bad at security operations.

If your answer to any of those questions was greater than zero, there is room for improvement. But there is no cavalry in the distance coming to miraculously address your operational issues. If anything SecOps is getting harder, for five reasons:

1. **Adversary innovation:** Adversaries are finding new ways to compromise devices, using both old and new tactics. They follow the path of least resistance to achieve their mission, with focus and persistence.
2. **Infrastructure complexity and dynamism:** With the advent of SaaS and the public cloud, technology infrastructure is getting more complicated, and changes happen much faster than before. Data ends up in environments you don't control and can't really monitor, but you still need to protect it.
3. **More devices, more places:** It seems every employee nowadays has multiple devices which need to connect to sensitive stuff, and they all want to access corporate systems from wherever they are. What could possibly go wrong? Compounding the issue is IoT and other embedded devices connecting to networks, dramatically increasing where you can be attacked. Maintaining visibility into and understanding of your attack surface and security posture continues to get harder.

4. **Hunters hunt:** This is a bit counterintuitive, but for a long time security folks could be blissfully unaware of the stuff they didn't find. If the security monitor missed it, what could they possibly do besides clean up the mess afterwards? Now organizations proactively look for signs of active adversaries in their environment, and these hunters are good at what they do. That creates a bunch of additional work — a good thing, because you are getting out ahead of imminent issues. But it does make SecOps harder by extending the list of stuff you don't have time to do.

5. **Skills gap:** We have been talking about a serious security skills gap for a long time, and it's not getting better. There just aren't enough security people to meet demand, and the problem gets worse each day.

Progress

But the news isn't all bad. By understanding the attacks which may be coming at you through more effective use of *threat intelligence,* you can benefit from the misfortune of others. You don't need to wait until you experience an attack and *then* configure your monitoring environment to look for it. Additionally, enhanced security analytics help you wade through all the noise to find patterns of attacks, and to pinpoint anomalous behavior which may indicate malicious activity.

Integrating threat intelligence with security analytics provides *Security Decision Support,* a key lever for scaling and improving the effectiveness of a security team. We will flesh these ideas out in detail in upcoming research.

But even with more actionable and better prioritized alerts, someone still needs to *do* something. You know — Security Operations. In many cases everything falls apart here. The security teams involved in many of the highest-profile breaches over the past few years were alerted to adversary activity more than once before attackers actually stole data. These companies just didn't execute sufficiently on a strategy to stop attacks before they became catastrophic.

> We have been talking about the need to Respond Faster and Better for more than a decade. As an industry we need to figure out how to more effectively operationalize world-class security practices, quickly and effectively.

Of course it's easy criticize organizations after a massive breach, but that's not our point. We bring them up as reminders of a concept we have been talking about for more than a decade: *Respond Faster and Better*. That's what it's all about. As an industry we need to figure out how to more effectively operationalize world-class security practices, quickly and effectively. And yes, we understand this is much easier to say than to do.

But *why* is it so hard? Let's examine what security operations teams tend to do with their time. Those of you with backgrounds in manufacturing probably remember time and motion studies which helped improve the productivity of factory workers. Security is far from a factory floor, but the concept applies. Can SecOps be streamlined by figuring out and optimizing whatever takes a lot of time?

We believe the answer is a resounding *yes*. A lot of security operational tasks involve updates, policy changes, compliance reporting, and other tedious rote work. Certainly there are periods of intense creative activity, such as triaging a new attack or trying to figure out an effective workaround. But plenty of time is spent on decidedly unsexy things.

This also creates unmet expectations for new entrants to the security field. Most new hires have dreams of being a l33t haXor or a threat hunter. Very few wake up excited to tackle change control for a list of firewall changes, or to reimage endpoints after the CEO clicked one of *those* links. Again.

And even if you could find people who get excited about day-to-day security operations, they would still be human. Which means they make errors. You need every update and change to be done right to avoid opening a hole in your environment large enough to drive a truck (or all your proprietary or customer data) through, so perfection is the goal — but people are not perfect, no matter how hard they try.

Behind the 8 Ball

So SecOps is inherently behind the 8-ball. The deck is stacked against us. Our attack surface is growing and our adversaries are getting better; we bring to the table our ingenuity, a metric crap-ton of alerts, and too few humans to get things done. It sounds like *Mission: Impossible*.

So what? Do we give up? Just pack it in and take a job at a coffee shop? To be honest, some days that sounds pretty good. Everybody loves coffee. But for folks who are passionate about security (like us), it's the wrong answer. We don't need to run. **But we do need to think differently.** We have to architect technology stacks smarter and more securely. We need to embrace automation instead of fearing it.

We are entering a new world. One where security is largely built into the technology stacks which run our infrastructure. Where we plan our operational functions and document them in clear runbooks. Where those runbooks are implemented via orchestration and automation within infrastructure without manual intervention.

This approach enables your security team to do what they are good at. They can understand the applications and design proper controls, evolve policies and tune the associated runbooks, and handle the exceptions which are inevitable in a dynamic environment. The machines take care of orchestrating all the different components of your environment to execute your automated runbooks. Then your people actually add value instead of just doing the same stuff over and over. That is the *Future of Security Operations*; this paper will dig into what that will look like, and how we believe you can get there.

To manage expectations, this future will require fundamental changes to how you do things, as well as embracing processes which will likely make you uncomfortable. As it should — major steps forward are uncomfortable for good reason.

Regaining Balance

Thinking differently about security entails taking a more enlightened approach, focusing the right resources on the right functions. We know it seems obvious that having expensive staff focused on rote and tedious functions is suboptimal. But most organizations do it anyway. We prefer to have our valuable, constrained, and usually highly skilled humans doing what humans are good at, such as:

- Identifying alert triggers that might indicate malicious activity
- Drilling into suspicious activity to understand the depth of attacks and assess potential damage
- Figuring out workarounds to address attacks

Humans in these roles generally know what to look for, but aren't very good at looking at huge amounts of data to discover patterns. Many don't like doing the same things over and over again — they get bored and become less effective. They don't like graveyard shifts, and they want work that teaches them new things and stretches their capabilities. They want to work in environments where they do cool stuff and can grow their skills. And — especially in security — they can choose where they work. If they don't get the right opportunity in your organization, they will find another which better suits their capabilities and work style.

On the other hand, machines have no problem working 24/7 and don't complain about boring tasks — at least not yet. They don't threaten to find another place to work, nor do they agitate for broader job responsibilities or better refreshments in the break room.

On the other hand, machines have no problem working 24/7 and don't complain about boring tasks — at least not yet. They don't threaten to find another place to work, nor do they agitate for broader job responsibilities or better refreshments in the break room. We're being a bit facetious here, and we don't advocate replacing your security team with robots. But in today's asymmetric environment, where you can't keep up with the task list, robots may be your only chance to regain some balance and keep pace.

It's worth expanding some concepts from our Intro to Threat Operations paper a bit, because over time we expect that vision of threat operations to become a subset of SecOps.

- **Enriching alerts:** The idea is to take an alert and add a bunch of information an analyst will likely want *before* sending it on to a human. This way they don't need to spend time gathering obviously relevant information from various systems and information sources, and can get right to work validating the alert and determining potential impact.

- **Incident response:** Once an alert has been validated, response generally includes a standard set of activities. Some activities can be automated via integration with affected systems (networks, endpoint management, SaaS, etc.), and responders can use the saved time to focus on higher-level tasks such as determining proliferation and assessing data loss.

Enriching Alerts

Let's dig into enriching alerts from your security monitoring systems, and how this can work without human intervention. We start with a couple different alerts and some educated guesses as to what would be useful to an analyst.

- **Alert: Connection to a known bad IP:** Let's say an alert fires for connectivity to a known bad IP address (thanks, threat intel!). With source and destination addresses, an analyst typically starts gathering basic information.

    1. *Identity:* Who uses this device? With a source IP it should be straightforward to see who the address is allocated to, and then what devices that person tends to use.

    2. *Target:* With a destination IP the external site comes into focus. An analyst would probably use geolocation to figure out where the IP is and a `whois` query to figure out who owns it. They could also find the hosting provider and search an intel service to see if the IP belongs to a known botnet, and then dig up any associated tactics.

    3. *Network traffic:* The analyst might also scan device traffic for strange patterns such as C&C or reconnaissance, or uncharacteristically large transfers to or from that device over the past few days.

    4. *Device hygiene:* The analyst needs details about the device, such as when it was last patched and whether it has a non-standard configuration.

    5. *Recent changes:* The analyst will probably be interested in software running on the device, and whether any programs have been installed or configurations changed recently.

- **Alert: Strange registry activity:** In this scenario an alert is triggered because a device has had its registry changed, but it cannot be traced back to authorized patches or software installation. The analyst would likely use similar information, but device hygiene and recent device changes would be of particular interest. The general flow of network traffic is also interesting, given that the device might have been receiving instructions or configuration changes from external devices. Registry changes alone might not be a concern, but much more suspicious just before or after a large inbound data transfer. Additionally, web traffic logs from the device could provide clues to what they were doing that might have resulted in compromise.

- **Alert: Large USB file transfer:** We also see the impact of enrichment in an insider threat scenario. Maybe an insider used their USB port for the first time recently, and transferred 1GB of data in a 3-hour window. That could generate a DLP alert, prompting someone to ask which internal data sources the device has been communicating with, and whether it has transmitted or received any anomalous data volumes over the past few days, which might indicate information mining in preparation for exfiltration. It would also help to review inbound connections and recent device changes, because the device could have been compromised by an external actor using a remote trojan.

In these scenarios, and another thousand we could concoct, all the information the analyst needs to get started is readily available within existing systems and security data/intelligence sources. Thus your enrichment process first orchestrates amongst all of these different information sources and then pre-populates the analysts tool in an automated fashion.

> All the information the analyst needs to get started is readily available within existing systems and security data/intelligence sources. Whatever tool an analyst uses to triage can be pre-populated with this information.

The ability to enrich alerts doesn't end there. If files are involved in the alert, the system could automatically poll an external file reputation service to see whether they are recognized as malicious. File samples could be set to a sandbox to report on what each one actually does, and whether it is associated with a known attack pattern or adversary. Additionally, if a file is identified as part of a malware kit, the system could then search for other related files, perhaps across other devices.

All this can be done before an analyst ever starts processing an alert. These simple examples illustrate the potential of orchestrated and automated enrichment to give analysts a chunk of what they need to figure out whether an alert is legitimate, and if so how much risk it poses.

Incident Response

Once an analyst validates an alert and performs an initial damage assessment, the incident is sent along to the response team. At this point a number of activities can be performed without a responder's direct involvement or attention to accelerate response. Potential responses to the alerts above nicely illustrate how orchestration and automation can make responders far more efficient and reduce risk.

- **Connection to known bad IP:** Let's say an analyst determines that a device connected to a known bad IP, meaning it could possibly be compromised and part of a botnet. What would a responder then want to do?

    1. *Isolate the device:* First the device should be isolated from the network and moved to a quarantine network with full packet capture to enable deeper monitoring and prevent further data exfiltration.

    2. *Forensic images:* The responder will need to take device images for further analysis and to maintain chain of custody.

    3. *Load forensics tools on the device image:* The standard set of forensic tools is then loaded up, and images connected for both disk and memory forensics.

All these functions can happen automatically once an alert is validated and escalated. The operations platform can connect via an API to the specific controls and devices to facilitate this. This kind of orchestration allows the responder to start with images from the compromised device, forensics tools ready to go, and a case file with all available information about the attack and potential adversary at their fingertips.

Opportunities to work faster and better don't end here. If the responder discovers a system file that has been changed on the compromised device, they can kick-off additional automated activities, orchestrating amongst their tools. They can search the security analytics system to see whether that file or a similar one has been downloaded to any other devices, run the file through a sandbox to observe its behavior and then search for matches, and (if they get hits on other potentially compromised devices) incorporate additional devices into the response process, isolating and imaging them automatically. This can accelerate the response by not sure assembling the information, but taking care of many activities once the analyst deems those need to be done.

> The key is the ability to accelerate SecOps by planning out activities in the form of runbooks, and then orchestrating and automating runbooks as executable security procedures to the greatest extent possible.

These techniques apply to pretty much any kind of alert or case that comes across a responder's desk. The registry alert above requires mostly memory forensics, but the same general processes apply.

Ditto for the large USB file transfer indicating an insider attack. But if you suspect an insider it's generally more prudent *not* to isolate the device, to avoid tipping them off. So that alert would trigger a different automated runbook, likely involving full packet capture of the device, analysis of file usage over the past 60-90 days, and notifying Human Resources and Legal of a potential malicious insider.

What is the common thread across all these scenarios? The ability to accelerate SecOps by planning out activities in the form of runbooks, and then orchestrating and automating runbooks as executable security procedures to the greatest extent possible.

Benefits

These seem self-evident, but let's review them anyway. This potential Future of Security Operations enables you to:

- **React Faster and Better:** Your analysts have better information because the alerts they receive include information they spend time gathering today. Your responders work better because they already have potentially compromised devices isolated and imaged; and a wealth of threat intel about what the attack might be, who is behind it, and a likely next move could be.
- **Operationalizing process:** Your best folks just know what to do, but other folks typically have no idea, so they stumble and meander through each incident; some figure it out alone, but others give up and find some other way to pay the bills. If you can have your best folks build runbooks which define proper processes for the most common situations, you can minimize performance variation and make *everyone* more productive.
- **Improve employee retention:** Employees who work in an environment where they can be successful, with the right tools to achieve their objectives, tend to stay. It's not about the money for most security folks — they want to do their jobs. If you have systems in place to keep humans doing what they are good at, and your competition (for staff) doesn't, it becomes increasingly hard for employees to leave. Some will choose to build a similar environment somewhere else — that's great, and how the industry improves overall. But many realize how hard it is, and what a step backwards it would be to manually do the work you have already automated.

So what are you waiting for? We never like to *sell past the close,* but we'll do it anyway. Enriching alerts and incident response are only the tip of the iceberg of SecOps processes which can be accelerated and improved with a dose of orchestration and automation.

Embracing the Machines

As we have explained in this paper, it is time to evolve Security Operations by leveraging technology to both accelerate human work and take over tedious rote tasks which don't add unique value. As we will illustrate through the rest of the paper, security orchestration and automation are terms you will hear pretty consistently from here on out.

Security practitioners have historically resisted the idea of automation, mostly because if done incorrectly the ramifications are severe and often career-limiting. So we advocate a slow and measured approach, starting with use cases which won't crater the infrastructure if something goes awry. We have discussed two of those in depth: enriching alerts and accelerating incident response.

Security practitioners have historically resisted the idea of automation, mostly because if done incorrectly the ramifications are severe and often career-limiting. So we advocate a slow and measured approach.

The value of being able to respond to more alerts, better and faster, is obvious. So we expect technologies focused on this constrained use case of Security Operations to become pervasive over the next 2-3 years. But the real leverage does not come from just making post-attack functions work better. The key question is: *How can you improve your security posture and make your environment more resilient by orchestrating and automating security controls?*

Before we dig into that we need some definitions of what automation of this sort looks like. And more importantly how you can establish trust in your automation. The Future of Security Operations depends on this. Without trust you are destined to remain in the hamster wheel of security pain (h/t Andy Jaquith). Attack, alert, respond, remediate, repeat. Obviously that hasn't worked too well, or we wouldn't continue having the same conversations year after year.

Orchestration-ready Tools

Constraints on the Future of Security Operations largely come down to technology and culture. Let's address technology first. You'll need most (eventually all) your controls to be accessible via API or other programmatic methods. Although we tend to get enamored with the automation aspect of advanced security operations, without the ability to orchestrate all the different tools in place… you are pretty much nowhere. Still stuck in the same vicious cycle, having analysts make changes in your control management consoles.

Orchestration-ready infrastructure and controls are not theoretical, rather close to a reality. It has been driven by (surprisingly enough) by Network Access Control (NAC) over the past decade. NAC forced the issue because enterprises needed a better way to reconfigure their network based on authorization, device security posture, and attacks in process. So early NAC vendors built hundreds of connectors to network and security devices to enable management and configuration of those devices.

Over time it made more sense for control vendors to open up more standard API to provide the same capabilities as the connectors offered (mostly) by NAC vendors. Orchestration today involves a mix of both proprietary connectors (built through technology alliances) and API. Moving forward we expect API to become prevalent in how we all manage vendor gear.

But for an organization needing to make all these tools work together, it doesn't really matter whether it's a proprietary connector or an API. Once you establish an environment where the Security Operations platform can manage the controls in place, you are ready to move to the next step: automation.

The Need for Trustable Automation

It's always interesting to broach the topic of security automation with folks who had negative experiences with early (typically network-centric) automation. They break out in hives when discussing automatically reconfiguring *anything*. We get it. When there is downtime or another adverse situation, ops people get fired and can't pay their mortgages. Survival instincts kick in, creating a cultural barrier and constraining use of automation.

Thus our focus on **Trustable Automation** – which means you tread carefully, building trust in both your automated processes and the decisions underlying them. *Iterate* your way to broader use of automation with a simple phased approach.

Thus our focus on Trustable Automation – which means you tread carefully, building trust in both your automated processes and the decisions underlying them. Iterate your way to broader use of automation with a simple phased approach.

1. **Human approval:** The first step is to insert a decision point into the process, where a human takes a look and ensures the proper functions will happen as a result of automation. This is basically putting a big red button in the middle of the process, giving an ops person the ability to perform a few checks and *then* hit it. It's faster but not really *fast*, because it still involves waiting on a human. Accept that *some* processes are so critical they will never get past human approval, because the organization just cannot risk a mistake.

2. **Automation with significant logging:** The next step is to take the training wheels off and let functions happen automatically, while making sure to log pretty much everything and have humans keep close tabs on it. Think of this as taking the training wheels off but staying within a few feet of the bike just in case it tips over. Or running an application in Debug

mode so you can watch exactly what is happening. If something does happen which you don't expect, you'll be right there to figure out what didn't work as expected and correct it. As you build trust in your process, we recommend you continue to scrutinize logs, even when things go perfectly. This helps you understand the frequency of change and which changes are made. You are developing a baseline of your automated process to will use in the next phase.

3. **Automation with guardrails:** Finally you reach a point where you don't need to step through every process. The machines are doing their jobs. Of course you still don't want things to go haywire. Now you can leverage your baseline. With your thresholds you can build guardrails to make sure nothing happens outside your tolerances. For example if you are automatically adding entries to an egress IP blacklist to block internal traffic to known bad locations, and all of a sudden traffic to your SaaS CRM system shows up on the queue for addition to your blacklist due to a faulty threat intel update, you can prevent the addition and alert administrators to investigate that threat intel update. This requires a deep understanding of the processes being automated and an ability to distinguish low-risk changes which should be made automatically from those which require human review. But that level of knowledge is what engenders trust, right?

Once you have built some trust in your automated process, you still want a safety net to make sure you don't go *splat* if something doesn't work as intended. The second requirement for trustable automation is *rollback*. You need to be able to quickly and easily return to a known good configuration. So when rolling out any kind of automation (whether via scripting or a platform), you want to make sure you store state information, and have the capability to reverse any changes quickly and completely. And yes, this is something to test extensively, both as you select an automation platform and once you start using it.

> Once you have built some trust in your automated process, you still want a safety net to make sure you don't go splat if something doesn't work as intended. You need to be able to quickly and easily return to a known good configuration.

The point is that as you design orchestration and automation functions, you have a lot of flexibility to embrace these concepts at your own pace. Some folks have a high threshold for pain and jump in with both feet, understanding that at some point they will likely need to clean up a mess. Others tiptoe toward this automated future, adding use cases gradually as they build comfort in the ability of their controls to work without human involvement. There is no right answer — you'll reach this orchestrated and automated future when you get there. But you *will* get there.

Given increasing trust in a more automated approach to SecOps, let's discuss additional use cases to illustrate the power of this approach.

Security Guardrails

We mentioned *guardrails* as one of the phases of building automation into your operational processes. Let's dig a little deeper into examples of how guardrails work within a security context. There are many other examples of putting guardrails around operations, network, and storage processes. But we're security folks so we'll discuss security guardrails.

- **Unauthorized privilege escalation:** Let's say you receive an alert of privilege escalation on a high-profile device (perhaps the CFO's phone). The trigger would be a log event of the escalation, which would result in rolling back the change and firing a high-priority alert at the SOC. If the change is legitimate you can always recommit. The CFO might be a bit miffed that your machines interrupted their work, but this kind of guardrail makes sure privileges remain as they should be unless the change is approved.
- **Rogue devices:** An unknown WiFi access point was detected using passive network scanning. It's not in your CMDB, as it would be if it went through your enterprise provisioning process, nor is it a type of device that your enterprise networking team would install, so it's safer to just take the device off the network until you can figure out why it's there and whether it's legitimate.
- **Deploy new IPS rules:** Finally, similar to the egress IP blacklist change above, IPS rules are automatically updated based on a trusted threat intel feed. But what happens if application traffic from your biggest customer is blocked because it looks like reconnaissance? In this case you can flag the customer's network as one that shouldn't ever be blocked and send a high-profile alert to investigate. Worst case, the block was legitimate (and the customer's network was compromised) — then you work with the customer to remedy *their* situation. To be clear, not automatically blocking the network opens a window of vulnerability on your network, but accepting that risk is a business decision.

These examples are all simple, but you can look at any runbook to find edge cases which would be problematic if bad changes happened automatically. Build guardrails for those scenarios, and then allow your machines to do their thing without threatening your environment.

Phishing Response

Another popular process for automation is handling phishing messages. Phishing is increasingly common, and it is resource-intensive to manually deal with every inbound message (shocking, right?). This is a perfect scenario for automation, which could look like this:

1. **Receive phishing message:** Your email security service flags a message as a phishing attempt and forwards it to a mailbox set up to trigger your automated process.
2. **Block egress:** Phish tend to travel in schools, so odds are good that similar messages will be sent to many of your users. So you take the message from the phishing mailbox, extract the URL, and then automatically update your DNS server to divert requests to that server to a safe internal address, which instead displays educational material about phishing.

3. **Investigate endpoint:** A user being targeted by a phish might be targeted by many other sketchy things as well, so you'll want to keep an eye on that device and automatically update your Endpoint Detection and Response (EDR) tool to increase logging frequency and depth. You'll also put the targeted employee on a watchlist in your SIEM/UBA product so they are subject to additional monitoring.

4. **Pay it forward:** You are not likely the only organization targeted by this phishing campaign, so you can automatically package up the information you got from analyzing the message and networking specifics, and forward them to your site takedown service. They will find the responsible ISP and initiate a request to take down the malicious site. Then folks less sophisticated than you can benefit as well.

You can also attach this phishing operational process to your incident response process. If your EDR information indicates a potential device compromise, you can automatically start capturing network traffic from that device and send it all to your response platform for investigation.

Exfiltration Response

We just talked about an inbound use case (phishing), so let's flip perspective to an exfiltration use case.

1. **DLP alert fires:** Unfortunately you probably get a number of DLP alerts every day — many are never investigated due to the volume of activity and lack of skilled resources to triage and investigate.

2. **Classify the issue:** You receive many different kinds of alerts, which require different responses (runbooks). For simplicity's sake let's say you consider the leak of account numbers or other personal data in email an inadvertent error, while an encrypted package going through the gateway is considered malicious.

3. **Kick off an educational process:** If the alert is deemed inadvertent you send a request to your security awareness training platform (via API) to register that user for a training module on protecting customer data. They can complete the training and be on their way without intervention by security personnel.

4. **Capture endpoint data:** If you determine the incident might be malicious, you immediately run a scan and then monitor the endpoint very closely. This process should also start assembling a case file and alert the SOC to a potential issue, described above under Incident Response.

5. **Quarantine device:** Depending on the results of your scan and telemetry analysis, if there is a concern of compromise you can automatically quarantine the device, pull images of memory and storage, and send a more urgent alert of an incident which requires investigation.

6. **Determine proliferation:** Once the type of attack is identified from the endpoint scan, you can automatically search existing endpoint security data to identify devices which were attacked similarly.

Almost this entire process can run in an automated fashion, leveraging logic and conditional tests to proceed appropriately. Depending on type an alert might kick off several different runbooks, each taking urgency and potential severity into account. Some organizations want human hands involved in the response process, so they establish interrupts for analyst review and possible intervention. For instance quarantine of endpoint devices might require approval by an analyst. The process is the same, except for an additional gate prior to quarantine and remediation, for manual approval.

You design your automated processes to work for your organization and its requirements. As mentioned above, you move toward full automation at a pace that works for you.

Updating SaaS Web Proxy

Finally, let's see how this approach works if you need to integrate with services which don't run on-premise. Many organizations have embraced SaaS-based secure web services, but some want more granular control over which sites and networks users can access. You might decide to supplement your service's built-in IP blacklist with multiple threat intelligence services to make sure you don't miss anything.

1. **Aggregate threat intel:** All your external data feeds can be aggregated in a threat intel platform (or your SIEM if you prefer), where you perform some normalization to see if any of several services identify a suspect IP address as bad.
2. **Block verified bad sites:** If an IP address shows up in multiple threat lists, it should obviously be blocked. But your SaaS service might already be blocking it, so you first poll your service for the IP's status. If it's already blocked do nothing. If it's not use the SaaS API to add the address to their blacklist.
3. **Monitor potentially bad sites:** For an IP showing up on just one list (meaning your suspicion has not yet been validated), you send an API request to the service to tighten policies for that IP. This likely entails more detailed logging, perhaps capturing packets to and from that device. Depending on the sophistication of your internal security team, you might also send them an alert to perform additional investigation on that IP for final determination.

This demonstrates the importance of API to automation. There is a logical flow, and the API enables clean integration between disparate services with higher-order logic.

Summary

This paper discussed a number of largely accepted use cases, including alert enrichment and incident response, as well as emerging use cases, to illustrate the value of orchestration and automation of your security operational functions, and the leverage available. To be clear, given the challenges of scaling security functions and meeting demand, we don't see any other way to achieve the security team's mission: to protect critical information. So we have no fear proclaiming orchestration and automation is the Future of Security Operations. You'll need to make sure your infrastructure of orchestration-ready, meaning that it can be managed by a third party, whether via connectors or more standard API. Then you automate where possible, supplement with internal resources as appropriate, and ultimately embrace these capabilities at whatever pace works for your organization.

Given the challenges of scaling security functions and meeting demand, we don't see any other way to achieve the security team's mission. Thus we have no fear proclaiming orchestration and automation is the Future of Security Operations.

But the core processes are similar regardless of how much degree of automation you embrace at any time. Without automation you just need to throw more people at it. You know, those people you can't find or retain. But we are all too aware of the role of trust in evolving toward this future. Without it you are stuck exactly as you are: likely understaffed, under-skilled, and falling short of expectations. Trust is not built overnight. It grows slowly, as you gain comfort in both the triggers that initiate your automated processes and the actions your processes take.

We recommend you tread carefully, first having humans ride shotgun on the process, approving each step. Then run without human intervention, but with detailed and granular logging to make sure you understand each step and action. Finally let the machine do its thing, with guardrails in place to ensure your process doesn't run amok and disrupt availability.

This is the future, whether you like it or not. So the sooner you start figuring out how to apply these tactics in your environment, the sooner you can give yourself (and your organization) a chance to keep pace with the attacks coming your way.

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 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/>.