HOW TO CREATE SECURITY PROCESSES THAT SOLVE PRACTICAL PROBLEMS

TABLE OF CONTENTS

INTRODUCTION

People and technology alone can't solve security problems efficiently, but when tied together with a process, they can be a powerful combo. While there are many processes that are so elaborate they end up holding back security teams, the best ones make it possible to get a job done faster, easier, and with more accuracy.

Processes, when done right, can:

- Increase the efficiency of your security team
- Align teams with tasks
- Align tasks with broader business goals
- Accelerate new hire training and knowledge sharing
- Scale an organization in any number of ways

The purpose of a process is to **solve real-world problems efficiently**. Every job function uses them, and if properly designed, benefits tremendously from them. Finance teams have a process for quarterly budgeting, DevOps teams are known for their continuous delivery processes, and HR teams have hiring and onboarding processes. Security can benefit just the same from processes, if not more, especially when it comes to responding to today's rapidly advancing threats.

WHY SECURITY NEEDS PROCESSES

Security is and has always been a complex job. **Connecting the work people do with the work that tools can do** is difficult without clear processes to tie them together. Processes enable cohesion across the security team to ensure nothing falls through the cracks, a critical benefit for teams that have to deal with a large volume of complex threats every day.

Processes can also help sustain a security organization over time. With the growing security-talent gap, competition to snag top talent is higher than ever. This means that not only is your security organization spread thin, but employees are often poached by whoever can offer the most money or the best benefits. This can especially hurt companies if the employees who jump ship leave without documenting knowledge of the effective strategies and workflows they employed during their tenure. This is where processes can save you from the downfalls of attrition.

Processes **enable the documentation of effective tasks and best practices**. Organizations that make process an ingrained part of their culture, whereby employees are incentivized to document and share effective strategies and workflows, feel the impact of attrition much less, ensuring such strategic knowledge can be put to use even as employees come and go.

For processes to be truly effective in solving real-world problems, they must be appropriately instated, well-documented, flexible, and easily shareable. To help your team meet these key criteria, we'd like to offer a framework for creating and implementing security processes.

This framework includes:

1. Starting with High-Level Business Goals
2. Determining Where You Need Process
3. Designing a Formalized Security Process

1. STARTING WITH HIGH-LEVEL BUSINESS GOALS

To solve real-world problems by way of a process, you first need to **define the ultimate goal.** Often, this is where processes go wrong, solving only for a single pain point without addressing the larger issue. While that approach can work for ad-hoc needs, for the purpose of this eBook, we will be focused on building the components of strong processes that stand the test of time.

To get clear on the goal of any given process, begin by defining:

The assets you're looking to protect

Your assets are what your adversaries are after, whether it's your point-of-sale (POS) system, a production-level password, or your servers. Start by defining which asset(s) you need to protect so you can ensure that every step in your process will help you get there.

Known threats

What are the biggest threats to the asset(s) you defined in the step above? It could be phishing if the asset is employee email accounts, credit card theft if the asset is your POS system, or even rogue employees if the asset is intellectual property.

Priority levels

Chances are, there will be a couple threats you could face for any given asset, so which one do you focus on first in the process? While all are important to protect against, start by ranking each threat on a scale of 1-10 (1 being low-risk, 10 being high-risk) so you can surface which threat is most critical to protect against first. You may also want to rank them based on likelihood. These numbers will help you as you define the steps in your process so you know how to prioritize actions.

Internal resources

Now that you know which assets you're protecting, what you're protecting them from, and the order of priority, ask yourself: what teams, tools, and budget do we need to to make the process come to life? As an example, you might need one security analyst, one security engineer, a security monitoring tool, and a malware protection tool to accomplish an alert escalation process. The resources needed will vary based on what your actual goals, assets, and threats are, so getting clear about your needs early on will help you both with budgeting and designing processes that work in the real world.

2. DETERMINING WHERE YOU NEED PROCESS

Not every security task warrants a process, so it's important to look at the goals you defined in step one and determine which ones can benefit from efficiency-driving processes and which ones are better off addressed ad-hoc.

To start, **understand where you can solve real-world, long-term problems.** Here are a few common types of processes that solve actual problems:

- Incident Response
- Vulnerability Management
- Application Security

How do you determine where your particular security organization needs processes? Start by evaluating the things your team already does on a regular basis, including:

Threat Hunting

Incident handlers typically sweep key servers for rogue listeners on an ad-hoc basis. Threat hunting is a prime example of a critical task that could be done far more efficiently, and with less manual overhead, if there was a process in place.

Let's look at how the Threat Hunting Project did this. After discovering a successful and preemptive way to catch undetected malware and policy violations, they turned their workflow into a formal process that now occurs on a scheduled basis and uses automated tooling to make it even faster.

App Scanning

Security-conscious developers often perform some form of application scanning. Using a static analysis tool, they can uncover security bugs before an app is deployed to production. Without a set process in place, though, app scanning is often an ad-hoc and manual task. (Which means it can easily slip through the cracks.)

With a process, app scans can be scheduled and automated, running at periodic times as part of the build pipeline and then reviewed by security engineers. This helps embed security into continuous deployment, a practice many DevOps-centric organizations are employing today to achieve better application security.

Phishing Investigations

When an employee reports a phishing attack, typically a senior member of the security team will triage the event. With phishing reports on the rise, it is no longer scalable for a security manager to manually review each event. This is another scenario where a process can enable scale and efficiency. With a documented process in place, investigations become repeatable by other members of the security team, relieving senior staff of tedious tasks so they can focus on more strategic decisions and on scaling processes across the entire team.

As you can imagine, each of the above processes will vary from company to company based on your goals, tools, personnel, budget, and more. That's why it's important to get clear on which processes you need and why.

Once you know where in your team's workflows it makes sense to implement repeatable security processes, it's important to know when to formalize and introduce one, with these being two major indicators:

Timing

Introducing a process too early can be dangerous if you don't know enough about its key components. And if your team is still small, you need to be careful not to weigh the team down by adding unnecessary red tape. But let things go on for too long without a process and tasks begin to slip between the cracks. There is a time and place for process, so weigh the pros and cons before implementing a process solely for the sake of formalization.

Scale

In general, a process is best introduced when it can address issues of scale, such as triaging 1000+ alerts a day. Manually handling these is far too inefficient and slow, so by introducing an automated process to investigate them, your security team can save time and move on to bigger things, while ensuring any important alerts are investigated.

Types of Processes to Consider

When thinking about processes, there are two main types: organic and designed.

Organic Processes

Organic processes are created as a direct response to a real-time need, often solving a short-term problem. They're typically quick and dirty, meaning they may not be the most efficient, and thus cannot sustain your security team long-term.

Designed Processes

Designed processes, on the other hand, are ones that are intentionally formalized up-front and intended to last for a long time. A good example is a SOC's process for triaging security alerts. Requiring multiple people and systems in order to triage accurately, alert handling is a prime candidate for a designed process to be put in place.

Designed processes, if not carefully crafted, can become over-designed, resulting in rigidity and inflexibility, thus slowing down productivity. Flexibility is extremely important in ensuring that processes can serve you across a broad range of scenarios and as your team grows.

There is a time and place for both organic and designed processes, so understanding when each is appropriate is important.

3. DESIGNING A FORMALIZED SECURITY PROCESS

Now that you know the criteria for where and when to introduce a formal process, it's time to learn how to create one. A good way to think about how a process works is that it's a lot like troubleshooting. There are a series of questions you should ask to get to the root of what you're looking to solve. Once you know what you're solving for, you can formulate a solution.

The questions you should ask include:

- What are we trying to solve?
  Example: "Automating the detection of phishing attacks."

- Is this a simple or complex problem?
  Example: "Complex. We have 1,000 employees distributed across the country using different browsers and operating systems, so new attacks are always throwing us for a loop."

- What is the scale of the problem?
  Example: "It has the potential to affect every single employee, so this is a big problem."

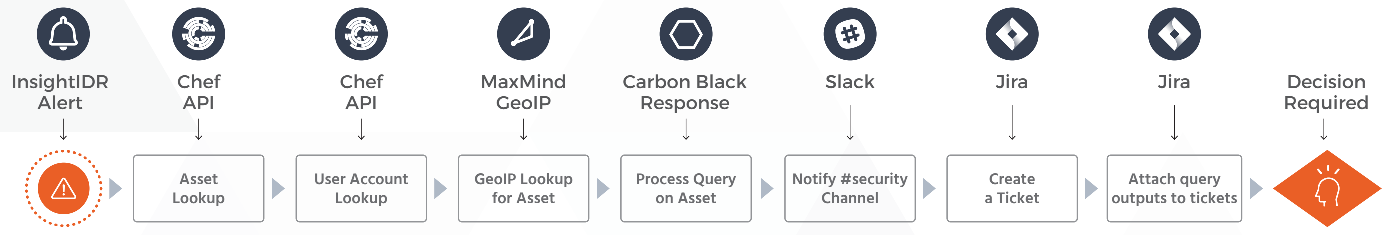
- What is the priority level of this problem?
  Example: "On a scale of 1 to 10 (1 being low priority, 10 being high priority), it's an 8. It can easily become an organization-wide issue, plus we're currently dealing with larger zero-day attacks and issues from unpatched software, which are also a high-priority."

- What information do we know?
  Example: "We know what operating systems are in use and that everyone is forced to change their email passwords every 90 days. We know what phishing attacks have hit us in the past and are prepared for them if they hit us again."

- What information do we not know?
  Example: "We don't know how many people are running outdated versions of their operating systems and internet browsers. We also don't know what we don't know. In other words, new attacks are cropping up daily. We also don't know how well-informed employees are about what a phishing attack looks like and how many times they've already fallen victim."

At this point, you should have a really clear picture of what you're looking to accomplish and who and what is involved. To document the components of the process, take your answers to the above questions and write out the process to respond using this formula to ensure everything is taken into account.

In a privilege investigation alert scenario, for example, when a user elevates privileges from a standard user to a system administrator (or 'root' user), it must be logged and reviewed by security personnel. The entire process might look something like this:



With a critical alert, several people on a team need to jump into action at once to accomplish all of these tasks. Moreover, there needs to be strategic human decision points throughout the process where team members can evaluate investigatory findings and determine what information to rule out as irrelevant and where to dig deeper with investigation.

Following a linear process in the privilege investigation example would be too rigid and time-consuming, and if there was in fact a true threat, this could be too much of a risk for your company. Instead, a parallel process whereby multiple people on the team can get involved simultaneously and similar tasks happen in cohesion can greatly reduce the time-to-response, increase the success rate of combating a threat, and optimize resources.

A Word of Caution About Linear Processes

When many people think about security processes, a step-by-step flow comes to mind, with one task followed by another. This is called a linear process. But processes don't have to be linear. In fact, we strongly advise against this. Not only does this train employees with the wrong mental model of defense, it ill-prepares you to respond to threats that morph and pivot.

We've written before on the dangers of linear thinking in security. Linear processes overlook the fact that attackers often pivot during their attacks, becoming highly unpredictable and able to circumvent most linear defenses altogether. That said, **the best approach is to design the steps in your process to run in parallel, or in graphs.**

Thinking in graphs means looking at where across your attack surface (including networks and endpoints) compromises could occur and how they could happen in parallel (rather than in succession). Thinking in terms of graphs enable responders to perform different investigatory tasks based on the real-time details of a compromise in tandem with an attack path.

This goes back to earlier points about flexibility: To be effective, processes must be able to adapt and evolve in the real world. After all, our adversaries usually don't follow a set path and are adept at changing their plans on the drop of a hat, so we as defenders should operate similarly.

4. TESTING A SECURITY PROCESS

Just as engineers run tests on their code before going live in a production environment, security processes should be tested before they are implemented in the real world.

Once you feel you've documented everything in the process you're designing, run though the instructions using a real use case. Ideally, the person who tests the process should not be the one who created it. Whichever use case you choose (vulnerability scanning, phishing detection, malware removal, etc.), be sure that it's one your organization will likely face, so you can be as prepared as possible when the time comes.

Mock up the use case and have your team follow the instructions you wrote out, accounting for the need to branch out and pivot at a moment's notice. They shouldn't follow the process by memory — see how the documented process works as it's spelled out out so you can note areas that need to be modified.

If you find that the process is not easy to follow, chances are it's either not defined well enough, or you don't have a solid understanding of the goal you are trying to accomplish. If that's the case, go back to the drawing board to revisit your goals, get an even better understanding of what the use case is, and flesh out your steps.

5. REASSESSING AND IMPROVING PROCESSES

As we mentioned earlier, to be effective, processes must be flexible. That means they not only should be able to adjust to each particular use case, but also evolve over time. Therefore, it's important to regularly reassess and improve your processes. In general, if there is a breakdown in the process, or if it isn't as efficient as it should be, the process should be reassessed.

The questions you and your team should ask during reevaluation include:

- Does the process accomplish the goals we defined?
- Is this process enabling efficiencies or slowing things down?
- Is the process still relevant to the threats we face?
- Do the benefits of the process outweigh the time required to follow it?

After your team conducts an honest evaluation of the process, it should be obvious whether the process is still functioning well.

If it is, great! That means you designed a flexible, long-lasting process that is serving your team well. If it's not, don't worry. Determine whether it's simply a matter of redesigning it to fit the nuances of a new use case or problem. That often happens. If so, perhaps you can reuse parts of the process. If you need to go back to the drawing board and create a net-new process, however, don't be afraid to do that.

Processes should also be reevaluated as your team grows. With new hands on deck, you want to be sure each team member has the opportunity to contribute and provide their expertise. The same can be said when new tools are added to your security toolbox.

Regularly evaluating your processes ensures that the work you put into them benefits your team and that you embrace an agile and adaptable approach to security, one that can give you a serious leg up on attackers who are simultaneously trying to do the same thing.

MARRYING PROCESS + AUTOMATION

Setting up processes isn't all sunshine and rainbows. Even if they provide efficiency gains in the long run, processes can still take quite a bit of manual time to implement. And after they are implemented, it's true that certain parts of a process can be set on auto-pilot (such as the scheduling of vulnerability scans, as we mentioned earlier), but many others will still require manual human intervention.

While manual processes can still introduce a great amount of efficiency just by defining a path to carry out a security task or respond to a threat, they can benefit security teams a great deal more when machine-to-machine automation is layered in.

Using security orchestration and automation, security teams can connect all their security tools together to create powerful processes, while still allowing for strategic human insight along the way. With orchestration and automation, you can move faster and better utilize your team's expertise for analysis and decisions, not manual investigation tasks. These operational gains can benefit your security posture and even help solve for the securitytalent gap.

At a time when security teams are constantly being asked to do more with less, automated processes offer a practical solution to get ahead and stay there, taking care of both routine and complex tasks with as little manual effort as possible.

ABOUT RAPID7

Rapid7 (NASDAQ:RPD) powers the practice of SecOps by delivering shared visibility, analytics, and automation that unites security, IT, and DevOps teams. The Rapid7 Insight platform empowers these teams to jointly manage and reduce risk, detect and contain attackers, and analyze and optimize operations. Rapid7 technology, services, and research drive vulnerability management, application security, incident detection and response, and log management for more than 7,100 organizations across more than 120 countries, including 55% of the Fortune 100.

ABOUT INSIGHTCONNECT

InsightConnect is a security orchestration and automation solution that enables your team to accelerate and streamline time-intensive processes—no code necessary. With 200+ plugins to connect your tools and easily customizable connect-and-go workflows, you'll free up your team to tackle other challenges, while still leveraging their expertise when it's most critical.