splunk>

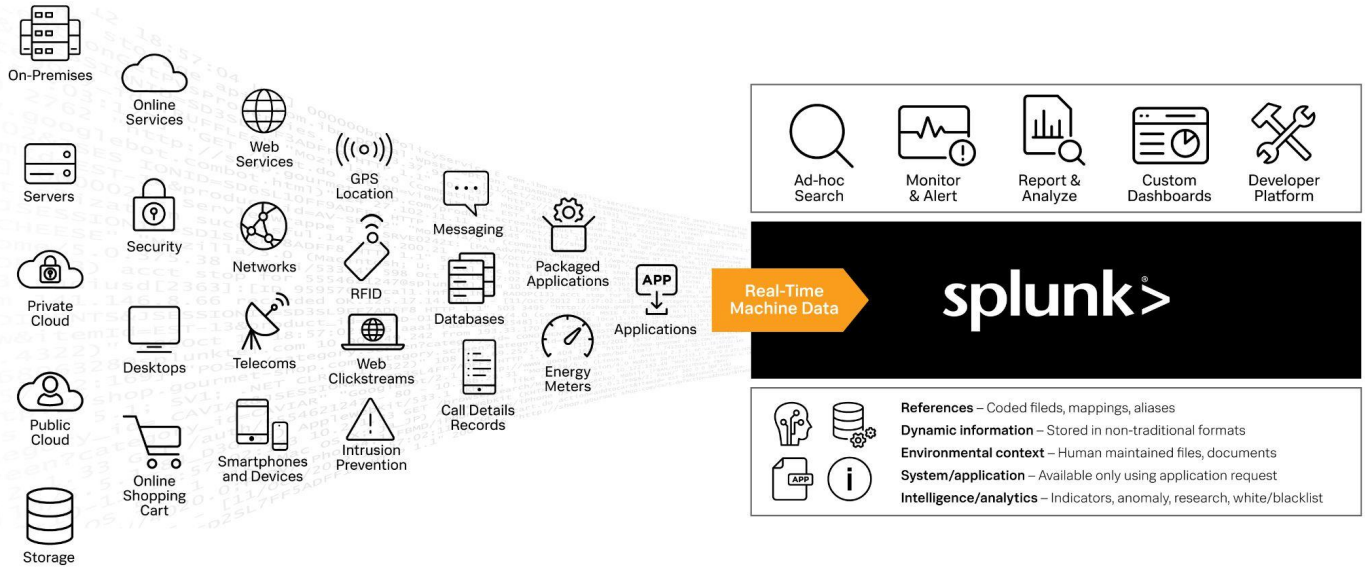
The Essential Guide to Risk-Based Alerting

Haylee Mills, Global Security Strategist

Contents

Chapter 1. Why RBA?

Security teams are drowning in data and overwhelmed with alerts. There must be a better way -- some esoteric or forbidden knowledge -- to produce high-fidelity alerts and keep your team from burning out.



An overview of Splunk's capabilities to turn data from any source into doing.

The good news is that there is a better way: Risk-Based Alerting (RBA). When Splunk customers use RBA, they see a 50% to 90% reduction in alerting volume, while the remaining alerts are higher fidelity, provide more context for analysis, and are more indicative of true security issues.

> "With risk-based alerting in Splunk Enterprise Security, investigations went from taking days to taking fifteen minutes, and our true positive rate has increased from 40% to 90% in under two months. We're discovering things that weren't possible to detect before." – Splunk Customer

The benefits of RBA include:

- A dramatic reduction in overall alert volume, producing higher-fidelity alerts.
- Improved detection of sophisticated threats like low-and-slow attacks that traditional security information and event management (SIEM) solutions miss.
- Seamless alignment with cybersecurity frameworks like MITRE ATT&CK, Lockheed Martin Cyber Kill Chain, CIS Critical Security Controls and National Institute of Standards and Technology (NIST).
- Increased analyst productivity, which means more time for high-value activities in your security organization.
- The ability to meet and exceed many security audit requirements results in a much smoother audit season.

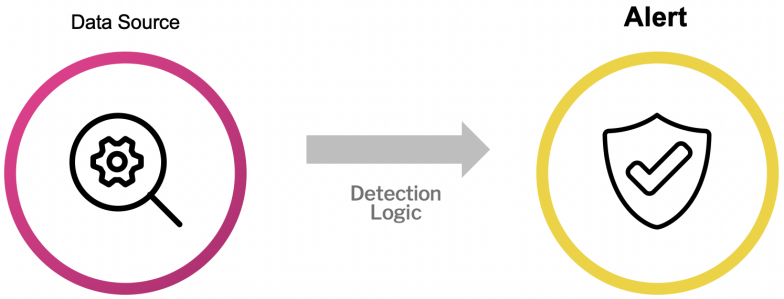
RBA provides teams with a unique opportunity to pivot cybersecurity resources from reactive to proactive while building out a flexible foundation to mature security operations across multiple departments. As alert fidelity and true positive rates increase, analysts are freed up to work on higher-value tasks, such as threat hunting, adversary simulation, or building up their skill sets and preparation to better face evolving threats.

How RBA Works

RBA uses the existing Splunk Enterprise Security (ES) correlation rule framework to collect interesting and potentially risky events into a single index with a shared language, which is then used for alerting. Events collected in the Risk Index produce a single Risk Notable only when certain criteria warranting an investigation are met, which means increased visibility and closing gaps while reducing the volume of low fidelity alerts. This process transforms traditional alerts into potentially interesting observations which correlate into a high-fidelity security story for analysts to investigate.

Traditional Alerting

In traditional cybersecurity alerting, there are one or more tools that forward data into a SIEM to detect potential issues and create alerts. The security team writes the detection logic or leverages prepackaged vendor content, alerting on suspicious activity that may be indicative of attacker behavior. It looks something like this:

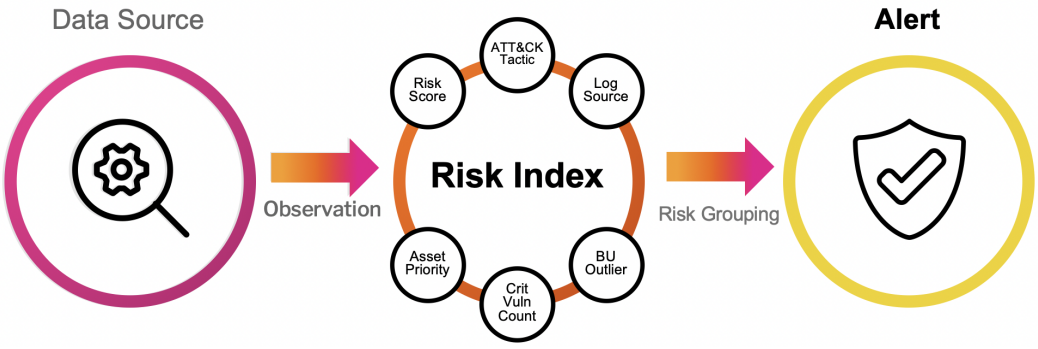


Unfortunately, alert volumes are overwhelming SOCs. More than 40% of organizations receive 10,000+ alerts per day, with 50%+ of those alerts turning out to be false positives. The sheer volume of alerts leads to abandoned or suppressed alerts, slower detection and response for true issues, and, yes, analyst burnout. I've been there, and that robotic response to endless benign events inevitably leads to situational numbness, which increases the risks of missing mission-critical alerts.

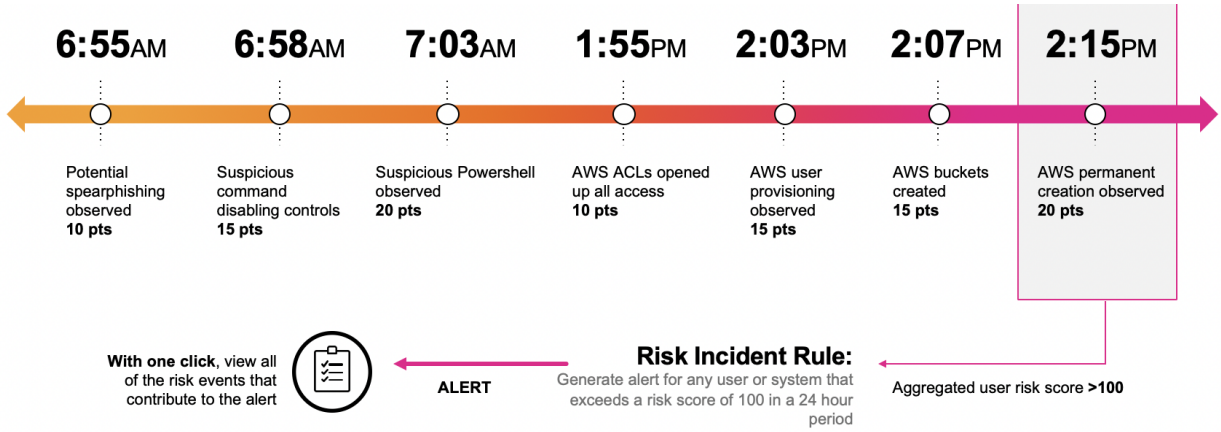
Let's look at traditional alerting from the viewpoint of a security analyst. The SIEM might generate three separate alerts, and each of those alerts is probably assigned to a different security analyst at a different point in time. Each analyst, seeing only one event in the string of activities, dismisses the notable. None of the analysts have the full context of the notable, so two things are happening here: each analyst gets too many low-level alerts which miss the bigger picture, and alert fatigue sets in as they burn out, working the same, mostly-benign, low-fidelity events.

Risk-Based Alerting

With RBA, Splunk's detection logic provides an observation, tags that observation with security metadata like alert source, ATT&CK technique, and a score, then dynamically modifies the risk score based on interesting attributes of the observed object, such as whether it involves a privileged user or an externally facing server. Alerting happens only when there are enough interesting observations correlated to the same object.



RBA doesn't send a Notable to an analyst until there are enough related observations in the Risk Index to trigger a higher-fidelity alert. A single analyst then sees all of these risk events collected in a single Notable and, with that context, can investigate faster and more efficiently to make a better decision about how to handle the alert. The diagram below shows an example of how a Risk Notable is generated by a Risk Incident Rule, using multiple events that would not individually trigger alerts.



What I love about RBA is how it can catch complex behavior over various time periods versus point-in-time alerting. For example, an impatient attacker might try various techniques on a single server over time. Here's three different alerts that use different factors and time scales to look for threat behavior:

- *Score Threshold 100 Exceeded Over 24 Hours* uses combined scores of events to trigger an alert.
- *Events from Multiple Sourcetypes Over 3 Days* generates an alert based on three unique data sources generating events from a single machine.
- *Multiple MITRE ATT&CK Tactics Observed Over 7 Days* uses observations tagged with MITRE ATT&CK tactics and techniques to create an alert.

Alert	Source	Tactic	Score
Non-standard Port Activity	Netflow	TA0011	10
Potential C2 Activity	Web	TA0011	25
Noisy IDS Alert	IDS	TA0001	15
New Registry Startup Key	EDR	TA0003	30
New Scheduled Task Created	EDR	TA0002	35

Using a variety of alerting criteria over variable time scales provides insight into regular behavior in your environment, as well as providing an insightful lens to use in tuning rules or in threat hunting.

Examples of How to Use RBA

We generally talk about RBA in terms of security data related to malicious compromise, but let's look at three different use cases that showcase the value of RBA in other contexts: machine learning, insider risk, and fraud. In each of these, the security domain encompasses fewer sources, meaning you can build a fully operational detection and response framework in less time, freeing up your team to work on bigger security issues.

Machine Learning

We've all heard tall tales of machine learning (ML) in some product that's "revolutionary" or "next-gen," but most of my data science pals seem busy attempting to set realistic expectations. There are so many unique problems that ML can complement or even solve single-handedly, but the challenge is in setting it up in the first place. How do you get useful information from the pile of security logs and other data that your organization collects minute by minute? In short, by alchemizing the artistry of domain-specific knowledge with the science of making computer programs do stuff with data. And that is where RBA can assist by generating datasets that emphasize context rather than noise.

Insider Risk

Compared to the massive MITRE ATT&CK framework and the menagerie of methods attackers can use to compromise machines, the amount of data sources and use cases needed to build out a valuable insider risk detection program with RBA is significantly reduced. Insider risk has a thematically similar amount of noisy alerts and data to sort through as in the cybersecurity context, but this enables you to build out a successful program in half the time.

Here are two excellent presentations about using RBA to manage insider risk:

- SEC1163A - Proactive Risk Based Alerting for Insider Threats - Matt Snyder from VMware discusses how they revolutionized their Insider Threat program with RBA, reducing investigation time from one-week pre-Splunk to 10 minutes with RBA.
- When Insiders ATT&CK! - At the MITRE ATT&CK conference, Matt digs into the details of building out Insider Risk detections and alerts with RBA and MITRE ATT&CK techniques for DLP/Exfiltration, Rogue Employee, or Oblivious User.

Fraud

Lastly, the Splunk App for Fraud Analytics leverages the RBA framework to alert on and investigate fraud. It helps address two pernicious fraud-related problems:

- Account Takeover (ATO) fraud
- Fraudulent activity seen from newly-registered accounts.

Researching events in its investigation dashboard can help you understand alerts and risk generation rules you can build as incidents and patterns are confirmed. After downloading this app, consult the User Guide to begin configuration for your environment. Implementing this app alongside an existing Splunk ES deployment takes some work, but it will save your team a ton of work after it is operational.

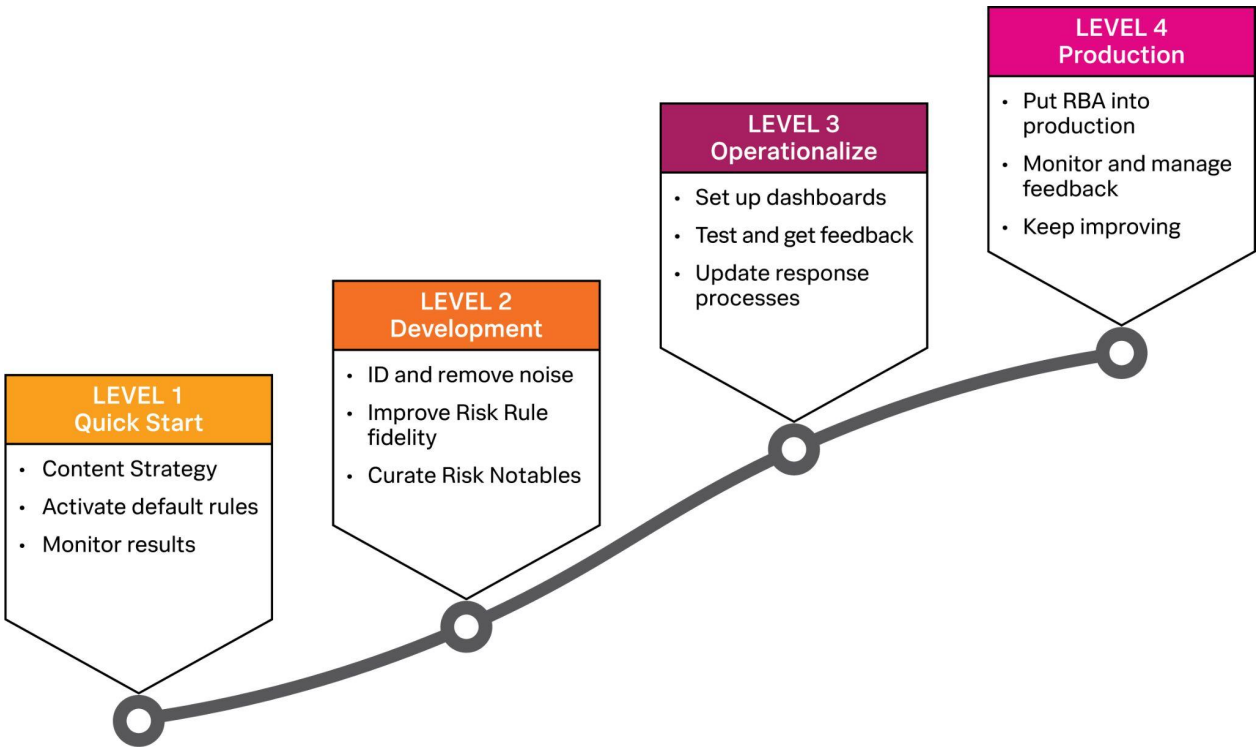
Chapter 2. The RBA Methodology

RBA is an iterative process, a maturity journey that lets you work your way up as time and resources permit. The best part is that after you've developed a strong foundation, it takes significantly less time to curate and develop new content to reach increasing levels of cybersecurity maturity.

Don't be scared! RBA is actually pretty exciting to use, especially once you start seeing your alert quality improve and alert quantity decline. Think of RBA as taking a lot of isolated security events and putting them together to tell better security stories with the context you need to make better, faster decisions. To put it another way, RBA is the magnet that pulls the big, scary security needles out of the alerting haystack.

The Four Levels

In working with our customers, the Splunk Superstar RBA Braintrust has developed a powerful methodology to kickstart your RBA implementation. From first moves to production, these four levels take you step-by-step through the process of successfully getting RBA up and running.



A four level approach to implementing RBA

Level 1 is all about getting familiar with how RBA works in your environment. It uses the defaults in Splunk ES to start with and then monitor and tune those rules to produce higher-fidelity alerts.

Level 2 is the classic development phase of any software-based project. You'll take what you learned in Level 1 to monitor and modify your existing rules to produce higher-fidelity Risk Notables.

Level 3 prepares your RBA implementation for production by setting up useful dashboards and modifying your existing case management processes to be more effective with RBA. In short, this level is all about getting RBA polished for real-world use.

Level 4 is the top of the mountain: time to Go-Live. Your team puts RBA into production and carefully monitors activity and results, fine-tuning rules and processes as needed.

RBA as a Project

As with any new process, RBA needs to be designed and implemented as a project within your organization, with scope, scale, and resources (people, machines, etc.), a dedicated project manager, and a timeline or schedule and tracking. At a minimum, the RBA project should include these actions:

- Get initial buy-in
- Set goals
- Build a plan
- Commit, track, and deliver

Getting Initial Buy-In

As an engineer, I often just want to build the thing and assume everyone will see the value immediately, but I highly recommend you develop buy-in at multiple levels before you begin your RBA implementation. It may take some effort to convince leadership that the time invested into building RBA enables them to meet or surpass many of their cherished security or resilience goals, but having them on board will make all the difference. My initial RBA blog post tries to cover this from a few different angles for various personas, and here are some presentations from Splunk customers using RBA who explain the value it has added to their organizations:

- SEC1249A - Accenture's Journey to Risk Based Alerting with Splunk Enterprise Security and Beyond - Chip Stearns from Splunk and Marcus Boyd from Accenture discuss how RBA transformed their SOC and the things to keep in mind while building it.
- SEC1803 - Modernize and Mature Your SOC with Risk-Based Alerting - Jim Apger from Splunk reviews RBA structure and benefits, then Jimi Mills from Texas Instruments offers a *detailed* timeline of TI's RBA evolution.

People at different vertical or lateral levels of the company have different goals on their radar, and knowing what they're trying to achieve may help you align with their goals. Decreasing the number of low-fidelity alerts while also decreasing the MTTR (Mean Time To Response) might be top of mind for a SOC director, while a CISO may be more interested in how security efforts map to a standard framework like MITRE ATT&CK. You can get in touch with a Splunk representative and ask for the Splunk Security Maturity Methodology (S2M2) Questionnaire with RBA to utilize a helpful spreadsheet for estimating the impact of RBA.

Important points to remember when discussing RBA within your organization:

- Leadership needs to understand the value-add of RBA so they will commit time and resources for it.
- Engineering requires time to build RBA as well as become familiar with the RBA methodology in production.
- Analysts need to be involved early on to become familiar with RBA investigations and co-develop responses and rules.
- Red/blue and/or purple teams can be valuable testers and partners for making changes or creating new rules.

Setting Goals

You can use RBA to improve different aspects of your SOC operations and security posture besides just covering visibility gaps with new content for alerting. We recommend that you select one or two goals and focus on those, along with devising metrics or means to measure success. Goals for your implementation might include:

- Increasing the quality of alerts so that analysts are more efficient when investigating security incidents. This may also feed into reducing response times because higher-risk events are prioritized.
- Using cybersecurity frameworks like MITRE ATT&CK, CIS Critical Security Controls, or the Lockheed Martin Cyber Kill Chain to improve the quality of alerts.
- Reducing the number of low-level alerts, especially those that are mostly "noise" and are dismissed without any investigation.

Longer-term goals might include:

- Enhancing and expanding the information stored in your Asset and Identity framework.
- Developing a risk library of metadata-enriched objects and behaviors for manual analysis or machine learning.
- Connecting to Splunk Threat Intelligence Management to leverage additional content sources and further improve risk evaluation and alerting.
- Expand traditional SIEM capabilities to new use cases and data sources, such as fraud or insider risk.

Building and Executing the Plan

For each goal you set, plan the actions or detailed steps needed to get the work done, measure progress, and assign owners to each action.

After you kick off the project plan, make sure you track progress, give regular status updates to stakeholders, and work through any roadblocks you may hit. While RBA can seem complicated at first, using the methodology in this guide will help you develop and execute a solid implementation plan.

Setting Expectations

Implementing a solid RBA strategy isn't a flick-the-switch solution, but it is foundational to improving your security maturity. While you can probably figure out how to do some kind of risk-based alerting with any security product, my goal with this guide is to share the proven RBA methodology that the Splunk team has developed while working with our customers to get you started on your own RBA journey.

Committing to RBA means investing in your people so they can transform your approach to cybersecurity. A successful implementation will change how your security team operates, empowering them to work on what matters and develop creative projects that leverage and synergize with RBA while relieving stress, workloads, and burnout.

All that being said, let's set some clear expectations for an RBA implementation:

- Initially, RBA may lead to more alerts. This sounds counter-intuitive, but it will actually help you improve the Risk Index and fine-tune alerts. We'll explain this in more detail later in the guide.
- RBA is an iterative process; it's not a one-time button you push to get results. It requires both analyst and engineer time to set up, fine-tune, and make ongoing changes over time. But trust us, it is worth it!
- RBA is not a magic solution to all of your alerting problems. You need to take the time to curate your Risk Index and understand the levers you can pull with RBA to customize it to your unique needs.
- RBA may challenge existing expectations of detection and response; do you really need to know about each detection within five minutes? If the answer is yes, you may need to solve performance challenges while building out your correlation search arsenal.
- To go along with that last point, your RBA implementation doesn't need to be 100% perfect before you start using it in production. There are small things you can do to achieve big results, or as I like to say, *lean into the 20% of effort that will give you 80% results.*

  Detection logic doesn't have to be perfect the first time around: craft the search and make sure the data returns, do some basic tuning of the noisiest, valueless noise, then throw that into the mix and see what happens! Especially before detections are in production, you will see how your new rule interacts with everything else and gain insight on what needs changing.

Your RBA journey will be worth the work, and I'll do my best in this guide to provide the methodology and tips that can help smooth that path. Typically, RBA users see anywhere from a 50% to 90% reduction in alerts, with the remaining alerts being higher fidelity and easier to investigate. That's definitely worth the work to implement in your organization!

How Long Is the RBA Journey?

How long will this RBA journey take? That's a good question! There's no one answer because every organization is different: different infrastructure, different resources and different security issues. Some teams will want to do everything on their own; others may choose to engage Splunk Professional Services or a Splunk Partner for guidance or to work alongside their internal security team.

Something I would like to make clear is that there is a decreasing amount of RBA-specific work as you progress through the different levels of the RBA methodology. How many resources you can dedicate to its development and the scope of your implementation also makes a huge difference. Once you set up the initial structure in Level 1, engineers are confident with developing, tuning, and tweaking in Level 2, and you've got process and response worked out in Level 3, then Level 4 is an open-ended maturation process that you can take any direction you choose.

One of our goals with this guide is to help you understand the work involved in taking the RBA journey so that you can decide which path is best for you. If your team has working knowledge of Splunk ES and core dependencies built out, then you will probably get through the first level in a week. If your team is just getting started with Splunk ES, it will take a bit longer and you may want to jumpstart your RBA implementation with some outside resources.

Chapter 3: Splunk ES and RBA

Before you dive into RBA, let's make sure you have the right Splunk essentials set up for the most effective RBA use and that you have a solid knowledge of the Splunk fundamentals used by RBA.

To use RBA as explained in this guide, you need to have Splunk Enterprise Security 6.4+ (ES) installed.

Within Splunk ES, RBA uses the following:

- Splunk ES Components:
    - Risk Analysis Adaptive Response Action
    - Notable Events
    - Asset and Identity Framework
    - Common Information Model (CIM) and Data Models
- Dashboards
    - Risk Analysis
    - Risk Event Timeline

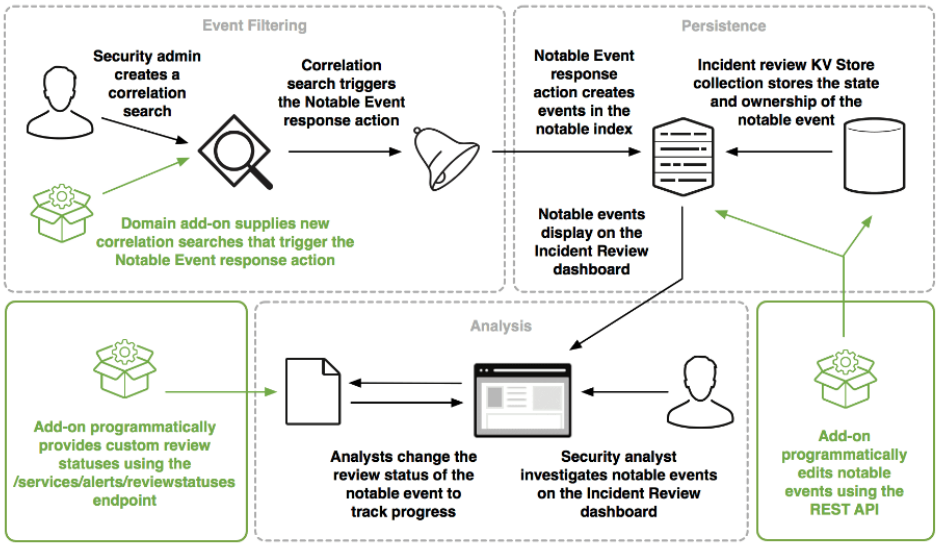
If you're new to Splunk or SPL (Search Processing Language), I have an Appendix carved out just for you. Check out the resources in Appendix A for where to get better at Splunk, at SPL, and even learning how to learn. It's some of my favorite content to share!

RBA Components

If you're not a Splunk ninja (or even close to it), this section will explain the Splunk components that RBA uses. It's important to know the function of each component and how they fit together, but it might make more sense as you follow along.

Terminology Test: If you don't understand the following **key terms**, please take the time to review the documentation links before you start working with RBA.

> RBA uses the existing Splunk ES **correlation rule framework** to collect interesting and potentially risky **events** in a single **index** from which to **alert**. These events produce a single Risk **Notable** when certain criteria are met.



Flow of a Notable Event created from events in Splunk into the Incident Review dashboard.

Risk Objects and Risk Scores

A risk score is a single metric that shows the relative risk of an object in the network environment over time. An object represents a system, an account, or any custom object you'd like to track and protect.

Risk Rules

A Risk Rule is a narrowly defined correlation search run against raw events to observe potentially malicious activity. A Risk Rule contains three components: search logic (Search Processing Language), Risk Annotations, and the Risk Analysis Adaptive Response action to generate risk events. All risk events are written to the Risk Index.

Examples of Risk Rules

- RR - Traffic to Non-standard Port
- RR - Threat Intel Match
- RR - Suspicious Logon Type

RBA Risk Rules can be sourced from the collection of out-of-the-box Splunk Enterprise Security rules from ES Content Updates (ESCU) or Splunk Security Essentials (SSE). Don't worry, we'll talk about prioritizing content development later in this guide!

Risk Notables and Risk Incident Rules (RIRs)

Risk Incident Rules (RIR) review events in the Risk Index and use an aggregation of events affecting a single risk object to create a Risk Notable. For example, a Risk Notable might be created when the RIR detects a single machine generating five risk events from various Risk Rules. They could have combined to cross a threshold of risk score, MITRE ATT&CK techniques, or unique data sources over various timeframes.

Splunk ES ships with two default Risk Incident Rules (correlation searches), which you can customize to identify threats based on risk in your security environment:

- ATT&CK Tactic Threshold Exceeded For Object Over Previous 7 days
- Risk Threshold Exceeded For Object Over 24 Hour Period

If you'd like to create custom Risk Notables, head to **Settings** → **Event types** then search for and edit `risk_notables` to include your own Risk Incident Rules. For details on the default Risk Incident Rules, see the Splunk documentation for Use Default Risk Rules in Splunk ES. If you'd like some other ideas, I've collected a few into Appendix B of this guide.

Risk Annotations

One of the coolest features of RBA is tagging your risk events with security metadata from common cybersecurity frameworks. By tagging your observations with these phases, you can catch attacker activity as they progress through the phases of an attack. Some of the frameworks available by default in correlation searches:

- MITRE ATT&CK
- CIS Critical Security Controls
- NIST
- Lockheed Martin Cyber Kill Chain

Even better, you can create your own custom framework. Just make sure to follow a naming convention so the events can be grouped together. For example, you could create a framework called *Potential Phishing* with three distinct phases on user activity that may indicate phishing:

- PDF Reader Spawns Web Browser
- User Traffic to Uncategorized Website
- HTTP POST to Uncategorized Website

You would then create a Risk Incident Rule for potential phishing when a user generates these three events from your custom *Potential Phishing* framework in a short timeframe. The possibilities are endless, and you can customize to whatever fits your organization.

Risk Scoring

The Risk Analysis Adaptive Response action applies a few key fields utilized in the Risk Analysis Framework:

- **Risk Message:** A unique message to describe the activity, which can use fields from the event surrounded by **"$"** such as: **"Suspicious Activity to $domain$"**
- **Risk Object:** What to track, observe, or protect and alert on. In most cases, this is a system or user.
- **Risk Object Type:** Can be defined as system, user, or something custom.
- **Risk Score:** The default score applied to these events, which can be modified by Risk Factors later on.
- **Threat Object:** I like to think of this as the "behavior" of a Risk Object, or what it is interacting with. I might choose **domain** from the Risk Message example to easily track the behavior of that domain across all risk objects.
- **Threat Object Type:** Some default examples include **domain**, **url**, **ip address**, **file hash**, **command line**, or **process name**. You can be creative and track anything you like.

As far as deciding on what score to use for each detection, I want to stress that you don't have to stress about this too much! Especially in the early stages, you'll see which events are scored far too high, and as you're learning to manage your risk ecology, it will be easier to decide on what needs to be scored higher.

The Threat Research team at Splunk uses a combined score based on Impact and Confidence multiplied and averaged together, which I really like, and we'll get into how to tweak that later in this guide. For example, within the detection Any PowerShell DownloadString, the team decided on an Impact of 80 and Confidence of 70, ending up with a Risk Score of 56. If you run that search historically before enabling it and find some regular activity in your environment, rather than lowering the score altogether, you can apply less risk to just those events; conversely, if you don't ever see that activity, you may decide to increase the risk score.

Risk Modification

The reason risk score is such a useful tool is that it is dynamic. Not only can you assign different amounts of risk based on what you see in the detection event – like zero risk when you see a failed HTTP POST and twenty for a success – you can also use contextual information about the risk object itself to tweak the score up and down.

For example, if you see commands like `systeminfo`, `ipconfig`, or `netstat` from a Field Services Technician user account on another user's computer, you would probably like to knock that sort of regular activity down. You can assign lower – or zero – risk to this type of activity, so you are still tracking this potentially malicious event that may at some point become part of a Risk Notable.

The way you do that within Splunk ES is with **Risk Factors** or **Ad-Hoc Risk Events**. Ad-Hoc Risk Events can be added to neutralize risk manually or as part of automation when closing an event, but Risk Factors are our dynamic buddies. Basically, you describe a field you want to look for, and if it contains a value you specify, then you can either add, subtract, or multiply the risk score at your discretion. This enables you to add additional risk for accounts with administrative privileges, executive machines, externally facing assets, and so on, or reduce the risk for known goods. You can even zero it out so it's being tracked but not alerted upon, then used with other contextual events to add risk only when seen together.

Asset and Identity Framework

Having a comprehensive set of Asset and Identity information for all your user and device objects is a key component for RBA success. When that information is available, you can use risk factors based on contextual asset or identity data. Splunk ES uses correlation searches to connect machine data with asset and identity data; RBA takes it to the next level by using those risk factors and customized risk scores to produce higher-fidelity alerts. At a minimum, the Asset and Identity Framework should include the following values for each asset:

- ip
- nt_host
- dns
- Category (optional)
- Priority (semi-optional)

Category and priority are specific to your organization and should focus on your specific risk profile. For example, you may want to mark C-suite laptops and logins as high-priority and IT people as a lower priority due to their different

roles within the organization. If you don’t have information for priority handy, just set a default for ES to use in its notable framework.

Every piece of information you can add to your assets or users is another useful lever you can tweak in Risk Rules. Knowing a user's business unit or a system's purpose helps guide analysts, and you can also use that information to adjust risk scores. The ability to automatically lower the risk of administrative process executions by business units without removing them entirely means you'll still track when admins perform their duties, but you won't inflate the risk score unnecessarily and flood analysts with alerts.

One of the most difficult problems to solve is mapping hostnames to IP addresses, especially with dynamic IP spaces in an increasingly remote-working world. The Splunk RBA team has seen a lot of success in bootstrapping Identity and Access enrichment with sources that contain hostname and active IP mapping, like proxy, authentication, VPN, or ZTNA logs. In some cases, EDR tools may have this data as well. No place has a perfect asset inventory, so try to think about alternative data sources you can use to help trace hostnames/users to IP addresses or integrate interesting metadata.

For more about the Asset and Identity framework, see the Splunk ES documentation for adding asset and identity data to Splunk Enterprise Security.

Common Information Model

Splunk organizes your data into a relevant schema through the Splunk Common Information Model (CIM). Then, it defines a Data Model (DM) to take relevant data from multiple sources such as Windows, Cloud, and VPN logs for the Authentication DM and make sure that they all use the same fields, such as src, user, and action, when looking at events.

When you make content, you can then use these Data Models to ensure you're looking across all of your relevant sources for detections. You don’t have to do this, but it is a best practice that will save you a lot of time in the long run if you set this up (or clean it up!) before you start building out your Risk Rules. I highly recommend Outpost Security’s Data Model Mechanic App for seeing what you need to adjust and where.

Also, don’t feel constrained by the CIM if your data has other useful fields you may want for new detections or that might be useful context for investigations. Adding another field to a Data Model is easy enough, and you can always slice up your logs with regex, calculations, or lookups to get exactly what you want in those fields. Keep in mind when editing default Data Models that an upgrade that affects those models will overwrite custom changes. You can either keep track of your unique fields when upgrading, create a custom DM that won't be affected, or update and verify your before and after data with this handy tool.

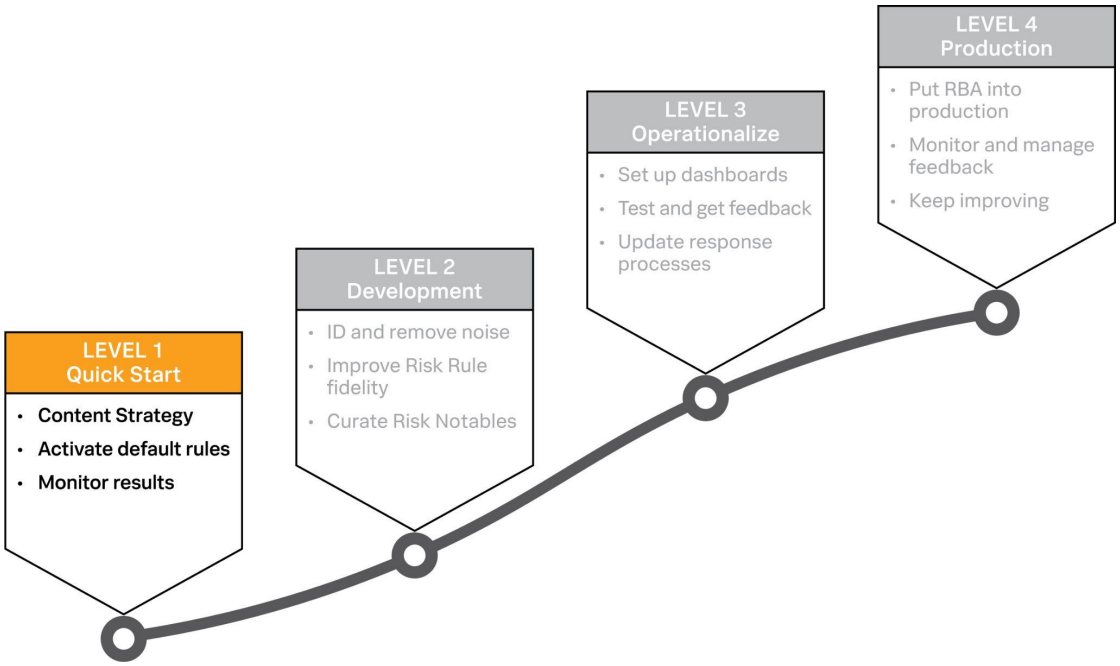
For more information on the Splunk CIM, read the Splunk documentation for Data Source Planning for Splunk ES.

It’s a little complex to get into this early in this guide, but Splunk utilizes something called Bloom filters against buckets of events to narrow down what it needs to search through to find the events you ask for. You can help Splunk do this 10-200 times faster by utilizing Accelerated Data Models (ADM) and the **tstats** command, which looks only at fields relevant to your Data Model. Using ADMs requires additional space on your search heads, and it takes getting used to tstats over raw search for your initial query. That being said, with all the content you can create with RBA, it’s a good idea to set an efficient foundation for your detections.

I highly recommend David Veuve’s tstats talk for more on this topic; even if it’s a bit technical; it gives a good idea of how and why it’s so powerful. Additionally, I want to make sure to mention that his Security Ninjutsu talks were life-changing for me as a security professional utilizing Splunk.

Chapter 4. Level 1: Quick Start

This first level of the RBA methodology eases you into using RBA with out-of-the-box Splunk ES features. Don't worry, you'll learn how things work as you follow this guide and become more familiar with how risk is generated in your environment.



Let's get RBA up and running in Level 1!

Goals

- Get started with RBA using standard Splunk ES features.
- Become familiar with monitoring Risk Rules and Risk Incident Rules.

Steps

1. Decide on initial content strategy and sources
2. Activate default rules in test mode
3. Monitor results

Step 1: Plan Content Strategy

If you're not sure exactly what to start building with RBA, there are a few great ways to approach your content development strategy. You'll want to pick about 10-20 Risk Rules to play around with in the first phase of implementation so you can see how they all interact and bubble up into Risk Notables.

Please keep in mind you don't have to do everything, everywhere, all at once. The ideas in this section are just suggestions to get you started.

One of the best-known content sources is MITRE ATT&CK, a widely-used knowledge base of adversary tactics and techniques based on real-world observations. Tactics are categories of activities (like Privilege Escalation or Command and Control), while techniques are specific activities (like Kerberoasting or Protocol Tunneling). Utilizing Splunk Security Essentials or Enterprise Security Content Updates (ESCU) can help you review which of your current data sources will cover which techniques, so that you can prioritize building out a breadth of detections across every tactic. Splunk ES is also equipped out-of-the-box to support NIST, CIS Critical Security Controls, and the Lockheed Martin Cyber Kill Chain, if any of those are used by your organization.

Here is a great presentation on using MITRE ATT&CK with RBA:

- **SEC1538 - Getting Started with Risk-Based Alerting and MITRE** - Bryan Turner reviews RBA structure and benefits, then guides building detections and aligning to ATT&CK.

One of my favorite places to start is whatever is wasting the most of your analysts' time. This really depends on the status labels you have set up in Splunk ES but go ahead and run the equivalent of this search (using your own labels) over the past three months:

```
'notable'
| stats count(eval(status_label="Incident")) as incident
count(eval(status_label="Resolved")) as closed
 BY source
| eval benign_rate = 1 - incident / (incident + closed)
| sort - benign_rate
```

You can also cut up the query into month by month or week by week chunks, which is useful when you've recently done serious tuning to identify which of your correlation searches are wasting the most cycles to investigate. Anything benign more than 50% of the time is a prime candidate for a better fit with RBA.

You probably can't take your traditional alerts out of production just yet, but go ahead and add a Risk Analysis Adaptive Response action with a risk score, risk message, risk object, and potentially a threat object to each of these correlation searches so that they'll start contributing to the Risk Index. Alternatively, you can clone the search and prefix with "`RR -` " to separate your new searches and revisit what's been tuned out. Traditional alerting typically has numerous exclusions to help reduce noise, which cause tunnel vision and limited visibility. As you work on tuning RBA rules once you understand how they work, you should be able to find patterns of `risk_object` attributes, certain fields in the event, or a `threat_object`, and then use those patterns to turn down scores for benign items and tune scores higher with reference data from confirmed incidents.

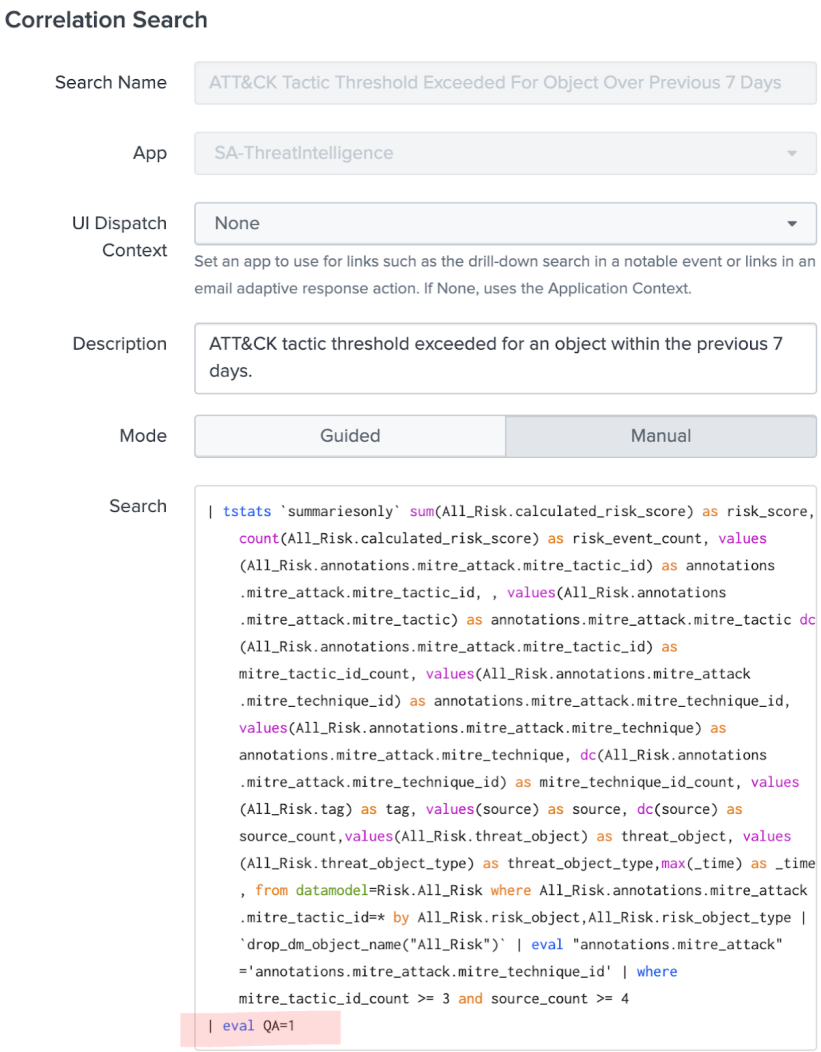
The final source I recommend is those data sources you just aren't getting enough value out of or where you've removed entire categories of alerts. This is really common with sources like EDR, IDS, and DLP, where they've got a ton of different detections out of the box, but you're only responding to High and Critical severity alerts because the volume is so high. You can copy the logic you have for ingesting those alerts and create a correlation search with the Risk Analysis Adaptive Response action for those lower severity detections and instantly have a ton of great content to plug into your Risk Index. These tools and other data sources are excellent for detections of subtle Living off the Land (Windows) or Break Out of Shell (macOS/Linux) techniques which would be impossible without something like RBA.

Step 2: Activate Default Risk Incident Rules

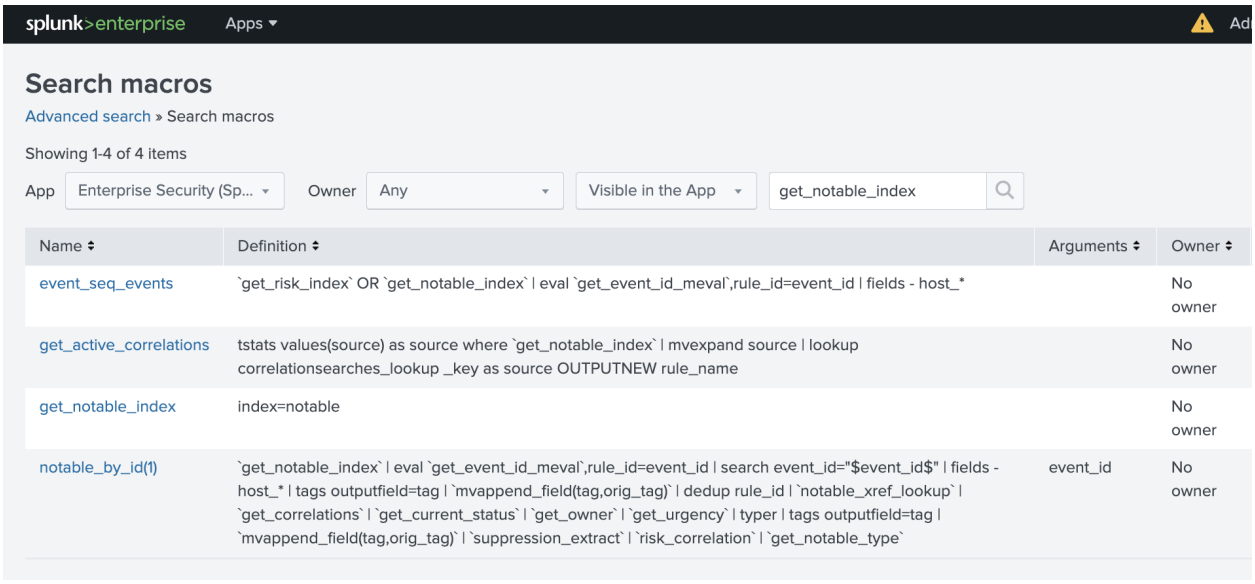
To carve out an environment for us to play around with risk without interrupting production workflows, let's create a test / QA mode for risk events and Risk Notables. We'll make some minor adjustments to the default Risk Incident Rules and then be ready to rock and roll with RBA.

1. Go to the Content Management dashboard (**Configure → Content → Content Management**) and check out the two default Risk Incident Rules:
   - **ATT&CK Tactic Threshold Exceeded For Object Over Previous 7 days.** This creates notables when the number of MITRE tactics exceeds three over the last seven days, i.e. `tactic_count >=3 AND source_count >=4`. This Risk Incident Rule searches the Risk Index for data diversity as defined by the MITRE ATT&CK framework. This rule is excellent for finding "low and slow" attacks.
   - **Risk Threshold Exceeded For Object Over 24 Hour Period.** This creates notables when the risk score for an object exceeds 100 over the last 24 hours, i.e. `risk_score_sum > 100`. It searches the Risk Index and aggregates risk scores by object. For example, if an object has eight related events, each with a calculated risk score, the search adds the eight scores together.
2. Add this line to the end of the SPL for each Risk Incident Rule:

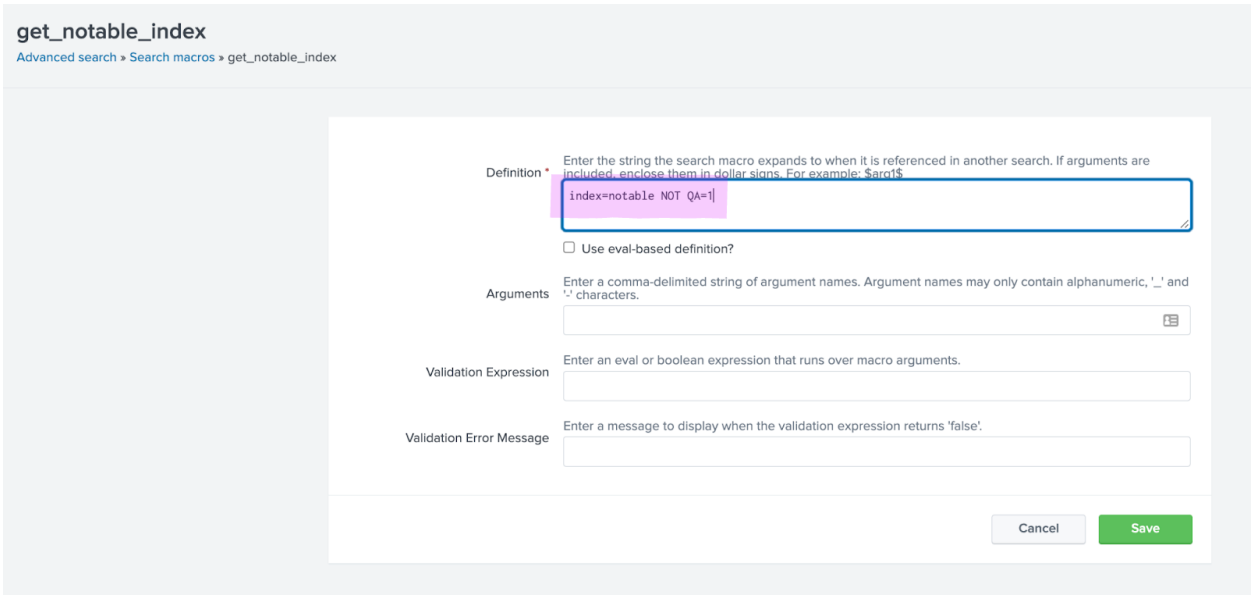
   ```
   | eval QA = 1
   ```



3. Go to **Settings** → **Advanced Search** → **Macros** and search for the `get_notable_index` macro:



4. This change populates the index that Splunk ES uses to alert, so let's make sure it doesn't alert on the new test notables:

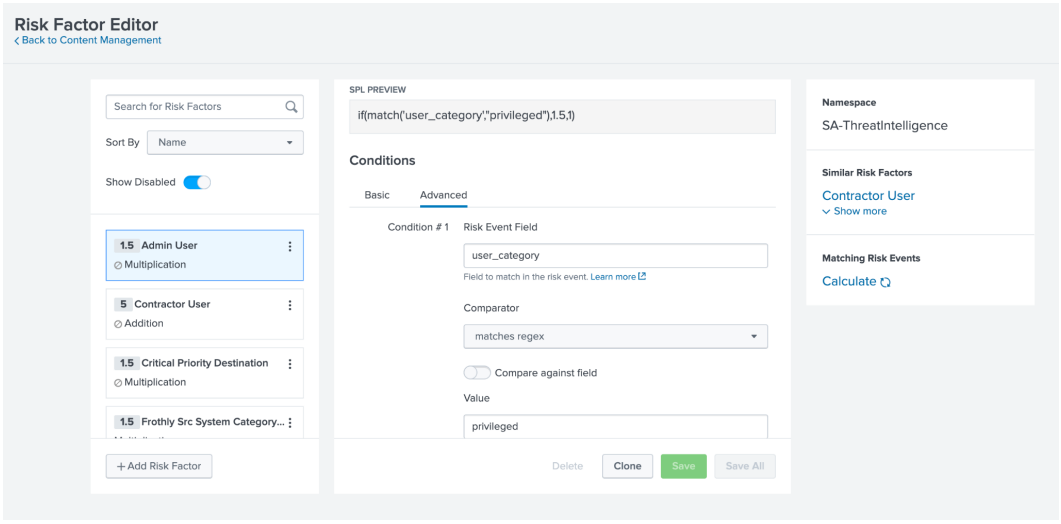


Great! Now the Risk Incident Rules fire in isolation while you test, monitor, and improve the Risk Index. Remember, this isn't all you can alert on; check out Appendix B for some additional Risk Incident Rule ideas.

Step 3: Enable Risk Factors

Risk Factors are one of my favorite features, and they're going to help make your dynamic score even more valuable. If you've got your Asset & Identity framework relatively well set up, you can enable some default Risk Factors out of the box using the Risk Factor Editor.

A super simple example would be bumping up the risk for users with admin or privileged rights. You can potentially enable this one out of the box if your Asset & Identity framework is populating the `user_category` with `privileged`:

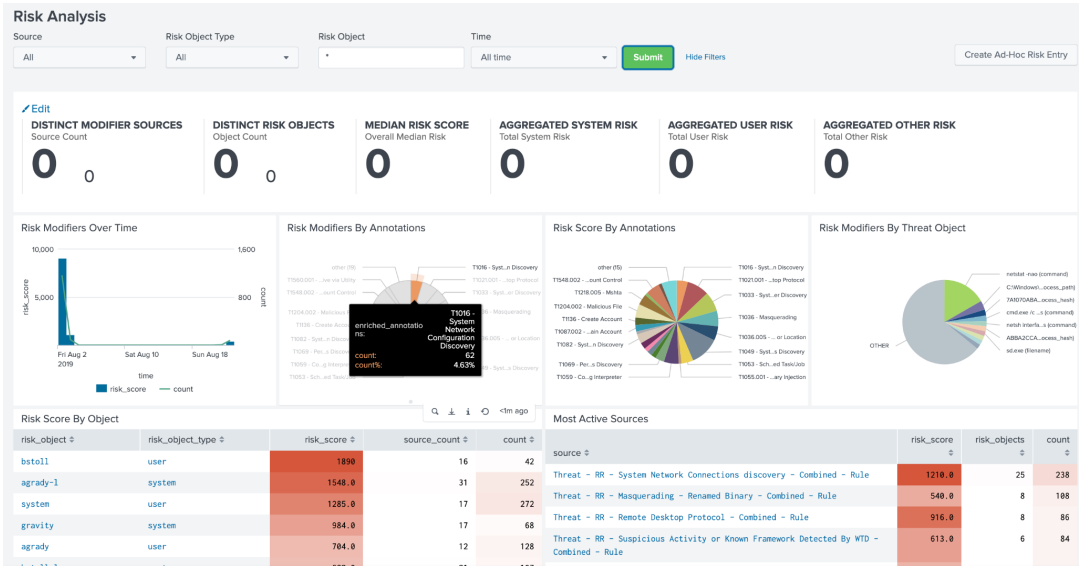


If you don't have exactly that (maybe you have `src_priority`, `src_category`, or some other useful field), go ahead and clone this default rule to match your data. If you've got some events firing in the Risk Index, now you can compare a privileged user to a regular user account and see how the score is multiplied by 1.5, or whatever you would like.

Step 4: Monitor Results

The Risk Index is a treasure trove of potential value and insight, and you're going to get really familiar with it as you build out RBA.

Just to get a brief idea of what's in there, head over to the Risk Analysis dashboard under the Security Intelligence menu in Splunk ES and you'll see something like this if risk has started populating:



Poke around and see what you can learn about what's happening in your environment, particularly the **Risk Score by Object** and **Most Active Sources** panels. Try plugging in a noisy risk object into the search field at the top, then view the **Recent Risk Modifiers** panel at the bottom to get an idea of what it's doing and what information is being relayed by the threat objects and risk message.

I've got all sorts of fun stuff to tell you about making risk scores more meaningful and Risk Notables more actionable in Level 2.

Extra Credit: Engage Red Teams, Go Purple!

Bringing in your red team to have an RBA-focused purple team engagement is an excellent source for Risk Rule development and content prioritization. Finding every trace of their activity in any log you have, developing Risk Rules from that investigation, then re-engaging with new content can show immediate value to leadership, engineers, and analysts.

As the RBA team begins to detect red team activities regularly, the red team must adapt and try increasingly subversive techniques that will highlight the need for additional data sources and protective controls to advocate. This back-and-forth rapidly identifies new detection priorities or makes clear what logs you really need in the SIEM to fill those gaps. As an added bonus, more interaction between your blue and red teams tends to create additional synergies in the future.

Level 1 Checkpoint

Risk Index is Populating

After you've enabled Risk Rules with the Risk Analysis action, there should be events being generated in:

```
index=risk
```

If that's not happening, there are a few reasons why:

- Incorrect SPL to generate events
- What the SPL is searching is empty
- Too many rules that don't fire very often

Don't be afraid to enable some rules that generate hundreds or thousands of events per day; this ain't your traditional old alerting! You want to have your events playing and interacting with each other. Enable some of those rules and then let's keep going.

Risk Notables are in Test Mode

A good way to make sure your sandbox mode works is to run:

```
index=notable source="*threshold exceeded*"
```

You should see the notables from the two default Risk Incident Rules and that the field "`QA`" exists in each of them. Then, you want to make sure that the notable macro is properly filtering them out. After that, you should be good to go.

Risk Factors are Applying

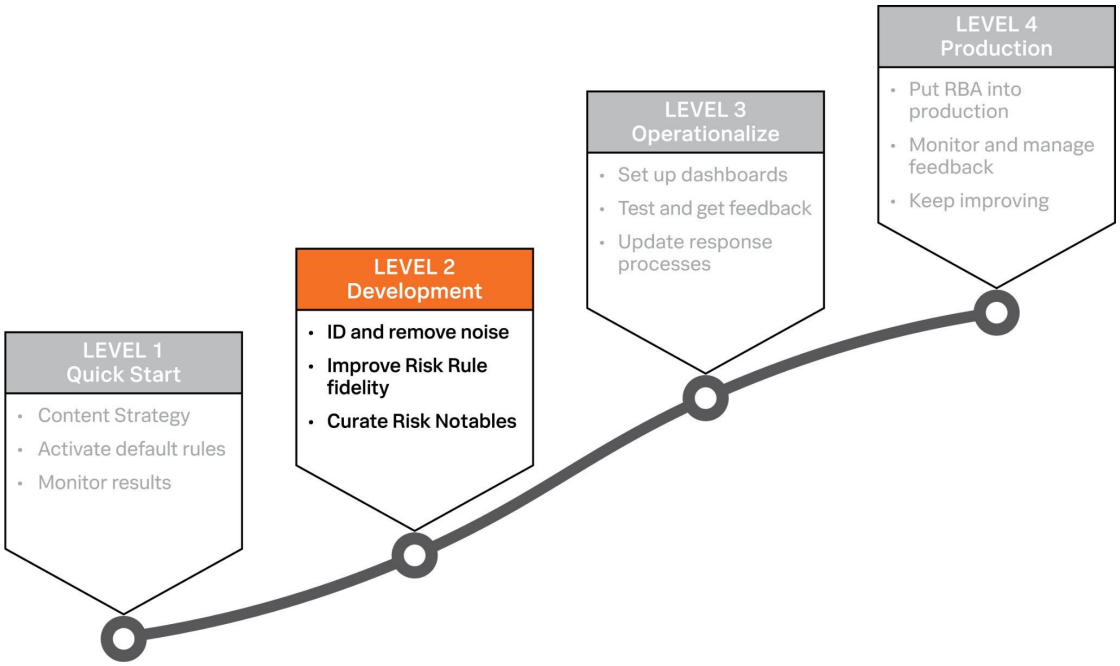
Look at the Risk Dashboard and plug in a system or account that should be modified by a Risk Factor. Compare this object to a standard object and ensure that the risk score is different. A quick reminder that Risk Factors are applied when the Risk Event is generated, so if you make changes, then you need to look at new events to see the change.

Lots of Noisy Notables

No problem! We'll look at different ways to adjust risk in Level 2.

Chapter 5. Level 2: Development

As you dive into RBA development, it's helpful to keep a mindset of try, learn, change, and move on. Not everything you try will show value right away, so be prepared to observe and study your Risk Index to learn about what is making a difference in your alert quality as you continue development.



Let's dive into the three Rs: Risk Index, Risk Rules, and Risk Notables!

A number of the techniques I mention in this chapter are covered in my 2022 Splunk .conf talk, SEC1144C - Curating your Risk Ecology: Making RBA Magick. Feel free to use that as a reference, although there are even more juicy bits in this guide.

Goals

- Learn how to identify and remove noise
- Improve Risk Rule fidelity
- Curate Risk Notables

Steps

1. Implement a lookup in a Risk Rule to remove noise
2. Implement a lookup in a Risk Rule to apply variable risk
3. Throttle a Risk Notable
4. Deduplicate events in a Risk Notable
5. Weight events for a Risk Notable
6. Create a QA mode for Risk Rules

Noise and Value in Risk Rules

Before building RBA, I was a security content engineer who did my darndest to make high-fidelity alerts. The reason I strove for this was because before that, I was a SOC analyst who worked way too many crappy alerts. When I had to build out content, I tried as hard as I could to monitor and allowlist regular business traffic and keep my analysts from responding to even more pointless volume as we attempted to close content gaps... it was a losing battle, to say the least.

My worst nightmare…

Obviously every environment is going to be different and what you're trying to do with RBA may be unique, but I find a lot of folks get stuck at some point in their implementation and feel like RBA isn't working as advertised. This is my worst nightmare! I just so happened to have a fertile combination of skills, experience, and ideas to see how RBA would make my life (and my analysts' lives) a million times better and I want everyone to feel that joy.

Usually those folks turn on some Risk Analysis actions, then risk starts generating, and soon things balloon into even MORE alerts than they were getting before. Or, risk objects start stacking up insane amounts of risk, without providing any clear value. If that's you, I get it! It isn't immediately clear how to best leverage this system to your advantage. Thankfully, RBA and SPL provide multiple ways to fine tune what you alert on and why.

So what is noise, exactly?

Well, as a content engineer, it's what isn't valuable in the detection context. Because RBA allows us to track observations in risk events, suddenly you are able to provide value to what was noise in a direct alerting pipeline, but I think it's worth having a critical eye for what is potentially valuable noise and what is nearly useless. The fun part for you is that it's different for everyone in every environment, so I want to make sure you have some methods to make the distinction.

One example of this…

Let's play with the idea of one detection: Windows MSHTA Child Process looks for the parent MSHTA (Microsoft's HTA Script Launcher) with some commonly exploitable child processes:

```
| where like(parent_process_name, "%\\\\mshta.exe") AND (process_name="powershell.exe" OR
process_name="cmd.exe" OR process_name="scrcons.exe" OR process_name="colorcpl.exe" OR
process_name="msbuild.exe" OR process_name="microsoft.workflow.compiler.exe" OR
process_name="searchprotocolhost.exe" OR process_name="cscript.exe"  OR
process_name="wscript.exe")
```

Let's say you want to deploy this, but you run this search on the past 24 hours and get hundreds of thousands of hits. Yikes! Rather than reject this outright, you can figure out what provides the most value.

Let's start by identifying fields to use to group events and find commonalities, so try searches like this with potentially relevant BY fields:

```
index=EDR parent_process_name="mshta.exe"
| stats count dc(src) as sources dc(users) as users
 BY cmdline
| sort - cmdline
```

Then, see if you can pick out command lines that clearly indicate regular business traffic to allowlist or tune down later. You want to use allowlisting as sparingly as possible, and only when the volume is so significant that it muddles the value you might get out of this detection. Let's take a noisy command line from the above search:

```
cmd.exe MaintenanceCheck.bat
```

This is from one service account performing regular activity on the server environment tens of thousands of times per day. You can create an allowlist for that specific combination of `cmdline`, `user`, and maybe even `src_category`, if you have reliable asset information. When adding to the allowlist, you want to do it in the most specific way possible, so that if something deviates from the regular pattern, you still observe it.

One more good idea…

As you work your way further down the list of noisy command lines, you see it isn't just one user. You drill down to discover these are service desk folks running a check on user machines when they call in with problems. Well, you could throw that into an allowlist, but if the volume is not substantial, you could bring the risk score to zero to prevent

these events from contributing to Risk Notables. The advantage is that if a Risk Notable alerts due to other suspicious traffic, analysts can see the service desk logging into the user workstation and understand the context around the other activity in the alert.

The idea is for us to ensure that events reaching our analysts are providing security value, but the other piece is ensuring our alerts do the same. Of course with RBA our alerts are made up of the events, so it's important to take a lens to our Risk Rules and Risk Notables. The whole purpose of this section is provide some context on how to approach allowlisting or tuning effectively before moving into the more technical, prescriptive suggestions in the next section. I highly recommend following along step by step in your own Splunk instance with your own data so you can see how the searches build up. Now, let's dive into some Risk Rules and SPL!

Step 1: Removing Noise with Lookups

When you are sure that you have some fields to specifically identify regular business traffic that won't be providing any value as risk, it's time to build out a lookup. I like to keep a tab open with the Lookup Editor App and add things I've found with searches like the ones in the previous section. I also like having a separate `justification` field, so if you build a dashboard for analysts to do the same we can track changes to a ticket for change management.

Once you have a lookup built out, it's pretty easy to insert it into a search like this:

```
index=proxy http_method="POST"
 NOT [| inputlookup RR_Proxy_Allowlist.csv
 | fields Web.src Web.dest
 | rename Web.* AS *]
```

You could also do this with a datamodel:

```
| tstats summariesonly=t values(Web.dest) as dest
 FROM datamodel Web.Web
 WHERE Web.http_method="POST"
 NOT [| inputlookup RR_Proxy_Allowlist.csv
 | fields Web.src Web.dest]
 BY _time,Web.src
```

Notice I'm specifying only the fields I want to include in my NOT statement, and with the index-based search I'm making sure to remove the Data Model prefix (`Web.`) required by a tstats search.

Step 2: Adjusting Risk

In most cases, you want to try and adjust risk rather than tune something out. First, let's look at how you could tweak scores for a particular combination of fields with an eval statement, then we'll try it with a lookup.

Using an Eval Statement

1. Start with a Risk Rule generating events from your proxy that is running antivirus detection, for example:

   ```
   index=proxy signature=*
   | table src user user_bunit dest signature http_code
   ```

   I've purposely brought along some fields that could be useful for making unique amounts of risk for various situations.

2. Now you can do something like this:

   ```
   index=proxy signature=*
   | table src user user_bunit dest signature http_code
   | eval risk_score = case(
   signature="JS:Adware.Lnkr.A","10",
   signature="Win32.Adware.YTDownloader","0",
   NOT http_code="200","25",
   signature="Trojan.Win32.Emotet" AND NOT user_bunit="THREAT INTELLIGENCE","100",
   true(),"50")
   ```

   Let's make a few distinctions here:

- Reducing risk to 10 for a particularly noisy signature or 25 when the connection was unsuccessful (`NOT http_code=200`).
- Zeroing risk for something noisy you want to track but not add risk.
- Adding risk when someone hits Emotet who isn't on the threat intelligence team testing malware
- Assigning 50 risk otherwise; the Risk Analysis action risk score will not apply if this field is created

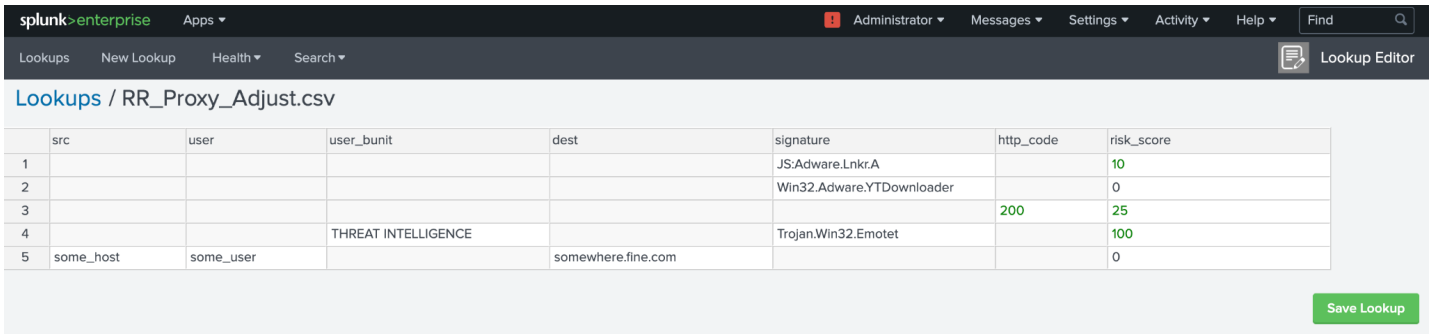
3. Slicing your results with `| stats count by useful_fields` means you can find relevant patterns for reducing or increasing risk. It may also be prudent to utilize Risk Factors to adjust based on the attributes of your risk objects.

Using a Lookup Statement

You could just as easily do this with a lookup instead of an eval statement:

```
index=proxy signature=*
| table src user user_bunit dest signature http_code
| lookup RR_Proxy_Adjust.csv src user user_bunit dest signature http_code
 OUTPUTNEW risk_score
| fillnull risk_score value="50"
```

Make sure that you include all of the fields to tweak in the lookup. It doesn't matter if they're blank when it isn't relevant. In this example, I threw in some new fields in case I find a particular `src` / `user` combo I want to adjust. A lookup which looks like this would work exactly like the eval statement, while being more accessible and easily editable:



	src	user	user_bunit	dest	signature	http_code	risk_score
1					JS:Adware.Lnkr.A		10
2					Win32.Adware.YTDownloader		0
3						200	25
4			THREAT INTELLIGENCE		Trojan.Win32.Emotet		100
5	some_host	some_user		somewhere.fine.com			0

Once you get used to using eval and lookups, you'll come up with all sorts of interesting ways to enhance your risk events and Risk Notables!

Identifying Noise in Risk Notables

Generating Risk Notables without curating them produces just as many notables as traditional alerting, so it's important to curate those notables early on. The goal here, as always with RBA, is to customize your notables to generate those higher-fidelity alerts that are actually meaningful to your security team. As you monitor and add more rules, items will "bubble up" and become more visible. Once these are spotted and adjusted, the signal to noise ratio will radically shift, alerts will decrease, fidelity of alerts will increase, and you'll be one step closer to production!

There are a few things I look at to see what might be contributing too much risk: correlation searches, threat objects, and Risk Notables themselves.

Correlation Searches

The “Most Active Sources” panel in the Risk Analysis dashboard is great for analyzing correlation searches generating risk. The SPL for this panel is:

```
| tstats summariesonly=false
sum(All_Risk.calculated_risk_score) as
risk_score,dc(All_Risk.risk_object) as risk_objects,count
 FROM datamodel=Risk.All_Risk
 WHERE * All_Risk.risk_object_type="*" (All_Risk.risk_object="*" OR risk_object="*")
 BY source
| sort 1000 - count,risk_score
```

Threat Objects

You can take one of the noisiest searches from that list and follow up with its threat objects:

```
| tstats sum(All_Risk.calculated_risk_score) as
score_sum,values(All_Risk.calculated_risk_score) as
score,dc(All_Risk.risk_object) as risk_objects,count
 FROM datamodel=Risk.All_Risk
 WHERE source="Threat - RR - Name of Our Noisy Rule - Rule"
 BY All_Risk.threat_object
| sort - score_sum
```

With both searches, you're looking for specific combinations of field values providing more risk than they should. This approach made a lot more sense as I made more Risk Rules, but an example might be a correlation search to ingest all EDR alerts making a lot of risk. A search like HTTP POST to *Potentially Suspicious Domain* should not be generating risk anywhere close to that amount for one alert versus the EDR’s hundreds or thousands. Either you’ll need to investigate regular results from that search and determine where to allowlist or tune, or adjust the risk score altogether. Perhaps both!

Speaking of threat objects, they’re also a great way to see what may be causing noise across all of your searches. I designed a search just for this purpose:

```
| tstats summariesonly=true count dc(All_Risk.risk_object) as
dc_objects dc(All_Risk.src) as dc_src dc(All_Risk.dest) as
dc_dest dc(All_Risk.user) as dc_users dc(All_Risk.user_bunit)
as dc_bunit sum(All_Risk.calculated_risk_score) as
risk_score values(source) as source
 FROM datamodel=Risk.All_Risk
 BY All_Risk.threat_object,All_Risk.threat_object_type
| `drop_dm_object_name("All_Risk")`
| sort 1000 - risk_score
```

This search may draw some additional insight into what needs your attention.

Risk Notables

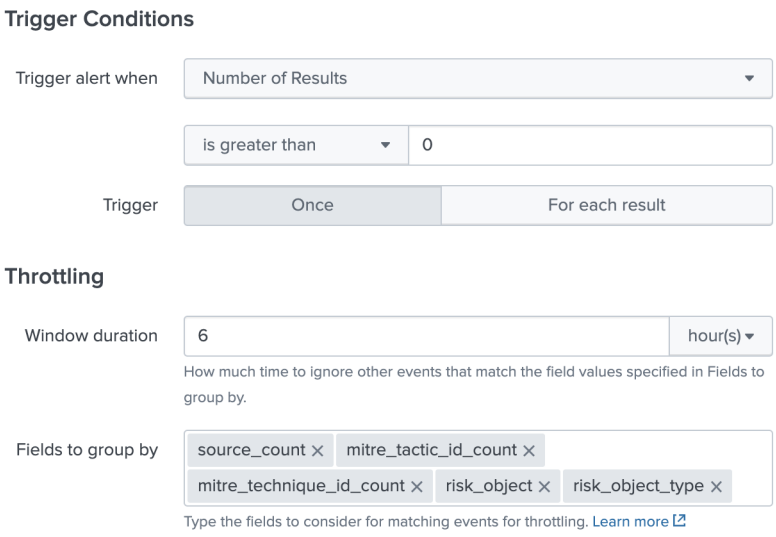
Lastly, look at the Risk Notables themselves and see if there are some correlation searches you may want to throttle, deduplicate, or weight properly. I'll talk later in this guide about how to do that. For now, try a search like this:

```
index=notable eventtype=risk_notables
| stats count
 BY orig_source
| eventstats sum(count) as total
| eval percentage = round((count / total) * 100,2)
| sort - percentage
```

This gives you an idea of which searches show up the most in your Risk Notables. Taking these high-level views with the Risk Index can help determine which Risk Rules you may want to tweak further, but you can also fiddle with the SPL of your Risk Notables to further reduce meaningless volume and generate more valuable alerts. Here we go!

Step 3: Throttling Risk Notables

When you first enable risk, you may see Risk Notables firing on the same risk object from additional risk events which shouldn't require additional investigation or even worse, Risk Notables which should have fired but didn't! Throttling compares the most recent search versus a previously defined timeframe to prevent an event matching all of those fields from firing again. These default *fields to group* by in the MITRE ATT&CK Risk Incident Rule ensures this looking over the past six hours:



However, maybe you have some useful data inside the risk_message you want to add to differentiate certain events. I might do something like this in the correlation search SPL:

```
| rex field="risk_message" "(?<risk_signature>.*) -.*"
```

This will pull out a new field called `risk_signature` from risk_messages like these:

- **EDR - Found Bad Thing** - *"bad_process.exe" on system_name1/user_name1*
- **DLP - Found Kinda Normal Thing** - *"file_name.csv" on system_name2/user_name2*
- **DLP - Found Kinda Normal Thing** - *"other_name.csv" on system_name2/user_name2*

If I added `risk_signature` to my throttle fields, notables now fire when the DLP or EDR found new events. Or, we could use this strategy to throttle something out. Handy!

Step 4: Deduplicating Events in Risk Notables

Another way to deal with stacking risk scores from similar events is to make sure each event is only counting once toward the total risk score. I explain this in my .conf22 talk, but let's break down a creative way of doing this in the Risk Notable SPL.

1. Start with the Risk Incident Rule's base search:

```
...
 BY All_Risk.risk_object,All_Risk.risk_object_type
| 'drop_dm_object_name("All_Risk")'
```

2. Add a `streamstats` command to retain the original score, sources, and messages:

```
...
 BY All_Risk.risk_object,All_Risk.risk_object_type
| 'drop_dm_object_name("All_Risk")'
| streamstats sum(risk_score) as original_score values(source) as sources
values(risk_message) as risk_messages
 BY risk_object
```

3. Now, add the wonderful `eval` command and `case()` to pick out some of the risk messages that tend to stack and inflate the score and leave the field `null()` if there is no match:

```
| eval adjust_score = case(
 source IN ("My Noisy Rule That Fires a Lot but I Still Want to Know About,
Once", "My Other Really Useful Context Low Risk Rule"),"1",
 match(risk_message,"IDS - Rule Category 1.*|IDS - Rule Category 2.*") OR
match(risk_message,"DLP - Rule Category 1.*|DLP - Rule Category 2.*"),"1",
 1=1,null())
```

4. Now you have a new field called `adjust_score` that you can use to combine these events if they match the criteria and use the `coalesce` function to take the new field (which just holds the value "1") if it exists, and if not, just use the `risk_message`:

```
...
| eval combine = coalesce(adjust_score,risk_message)
```

5. Now, `dedup` on that new `combine` field and sum the results with the saved fields from `streamstats` to have these noisy alerts we've defined count only once:

```
...
| dedup combine risk_score
| streamstats sum(risk_score) as risk_score values(sources) as source
values(risk_messages) as risk_message
 BY risk_object
```

Here's all that code in one place:

```
...
 BY All_Risk.risk_object,All_Risk.risk_object_type
| `drop_dm_object_name("All_Risk")`
| streamstats sum(risk_score) as original_score values(source)
as sources values(risk_message) as risk_messages
 BY risk_object
| eval adjust_score = case(
 source IN ("My Noisy Rule That Fires a Lot but I Still Want to Know About, Once", "My
Other Really Useful Context Low Risk Rule"),"1",
 match(risk_message,"IDS - Rule Category 1.*|IDS - Rule Category 2.*") OR
match(risk_message,"DLP - Rule Category 1.*|DLP - Rule Category 2.*"),"1",
 1=1,null())
| eval combine = coalesce(adjust_score,risk_message)
 | dedup combine risk_score
| streamstats sum(risk_score) as risk_score values(sources) as source
values(risk_messages) as risk_message
 BY risk_object
...
```

Make sure to retain any necessary fields for the Risk Incident Rule logic to function. You definitely want to keep these observations in the Risk Index un-duplicated in case you need to research and investigate, but this will keep certain events from over-alerting; you don't want analysts to investigate a Risk Notable that was technically already handled. Using this technique to manage contextual, observational, noisy events can greatly reduce alert volumes and ensure they are still contributing to Risk Notables. Another great method by David Dorsey for capping how much risk a single source can contribute is on the RBA Github.

Step 5: Weighting Events in Risk Notables

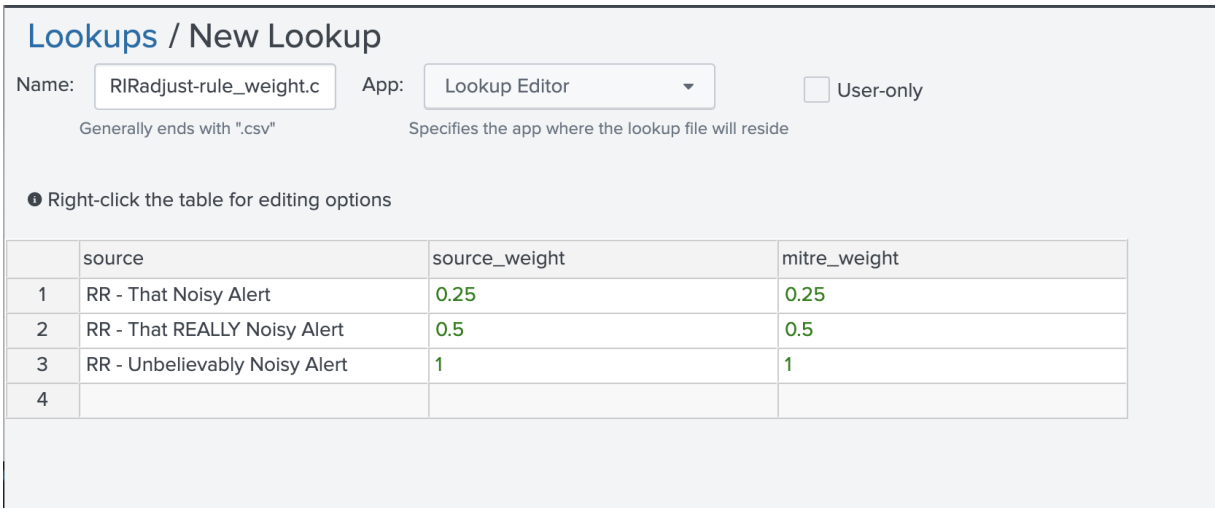
As you enable more rules with low or zero risk scores, you might find them over-contributing to Risk Notables that count the number of sources or MITRE ATT&CK tactics. A convenient way to handle this is by weighting how much they actually contribute to those counts.

1. Let's start by tacking on some logic to the SPL of ATT&CK *Tactic Threshold Exceeded For Object Over Previous 7 Days*:

   ```
   ...
    BY All_Risk.risk_object,All_Risk.risk_object_type
   | `drop_dm_object_name("All_Risk")`
   | mvexpand source
   ```

   `mvexpand` creates individual events from what is in the multi-valued source field, so you can apply distinct weights to various rules before combining them back together.

2. Craft a lookup where the weight field will be subtracted from 1, so you have something like this:



	source	source_weight	mitre_weight
1	RR - That Noisy Alert	0.25	0.25
2	RR - That REALLY Noisy Alert	0.5	0.5
3	RR - Unbelievably Noisy Alert	1	1
4			

3. Add this into the lookup logic:

   ```
   | lookup RIRadjust-rule_weight.csv source
    OUTPUTNEW mitre_weight source_weight
   | eval mitre_weight = if(isnotnull(mitre_weight),mitre_weight,"0")
   | eval source_weight = if(isnotnull(source_weight),source_weight,"0")
   ```

   So, if a rule is in the lookup, the weight is reduced from 1, otherwise it makes every other rule subtract nothing from the total count.

4. Combine the events back together and subtract from the current count for sources and MITRE tactics:

```
| streamstats sum(mitre_weight) as mitre_weight_total sum(source_weight) as
source_weight_total values(*) as *
 BY risk_object risk_object_type
| eval mitre_tactic_id_count = mitre_tactic_id_count - mitre_weight_total
| eval source_count = source_count - source_weight_total
```

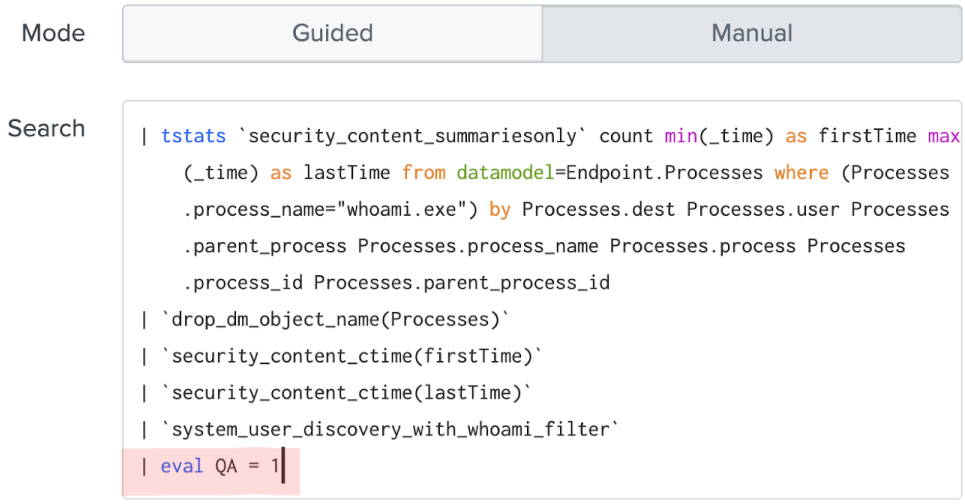
Now we're in business! Those noisy but useful rules you tuned the risk score down for can be tweaked in the Risk Incident Rules utilizing other security metadata. The SPL after the BY clause of the original search should now look like this:

```
...
BY All_Risk.risk_object,All_Risk.risk_object_type
| `drop_dm_object_name("All_Risk")`
| mvexpand source
| lookup RIRadjust-rule_weight.csv source
 OUTPUTNEW mitre_weight source_weight
| eval mitre_weight = if(isnotnull(mitre_weight),mitre_weight,"0")
| eval source_weight = if(isnotnull(source_weight),source_weight,"0")
| streamstats sum(mitre_weight) as mitre_weight_total sum(source_weight) as
source_weight_total values(*) as *
 BY risk_object risk_object_type
| eval mitre_tactic_id_count = mitre_tactic_id_count - mitre_weight_total
| eval source_count = source_count - source_weight_total
| eval "annotations.mitre_attack" = 'annotations.mitre_attack.mitre_technique_id'
| where mitre_tactic_id_count >= 3 and source_count >= 4
```

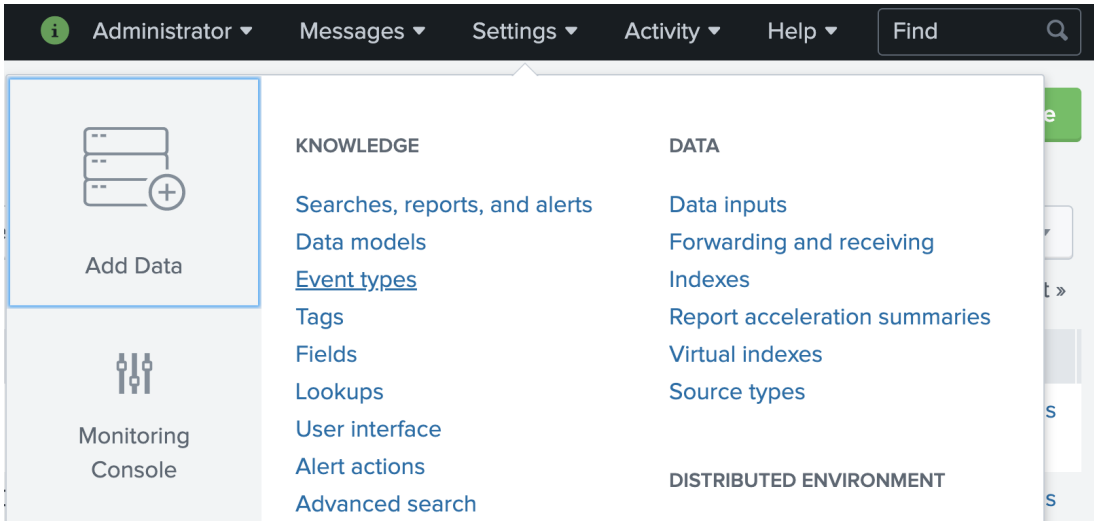
Step 6: Creating a Risk Rule QA Mode

Before you put an RBA rule into production, it's important to try it out in test / QA mode. In building out your risk ecology, new rules and variables can react strangely and suddenly. QA mode, then, provides a sandbox where you can compare how many Risk Notables you might have before and after you implement a batch of new content. Let's get that QA mode set up right now.

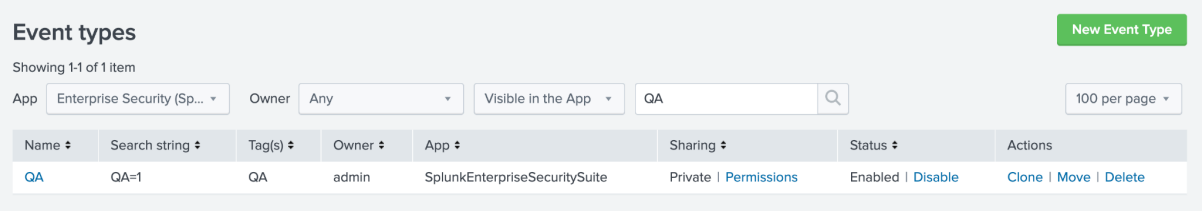
1. Add `eval QA="1"` to the end of the new Risk Rule being tested:



2. Add an exclusion to the Risk Incident Rules so the events labeled this way won't be counted. There are a few ways to do this, but because these Risk Incident Rules are utilizing the Risk Analysis Data Model, you need to make sure that whatever you do comes through the Data Model. You do that using the already existing `All_Risk.tag` field and creating a new eventtype to add the tag.
    a. Go to **Settings** → **Event types**:

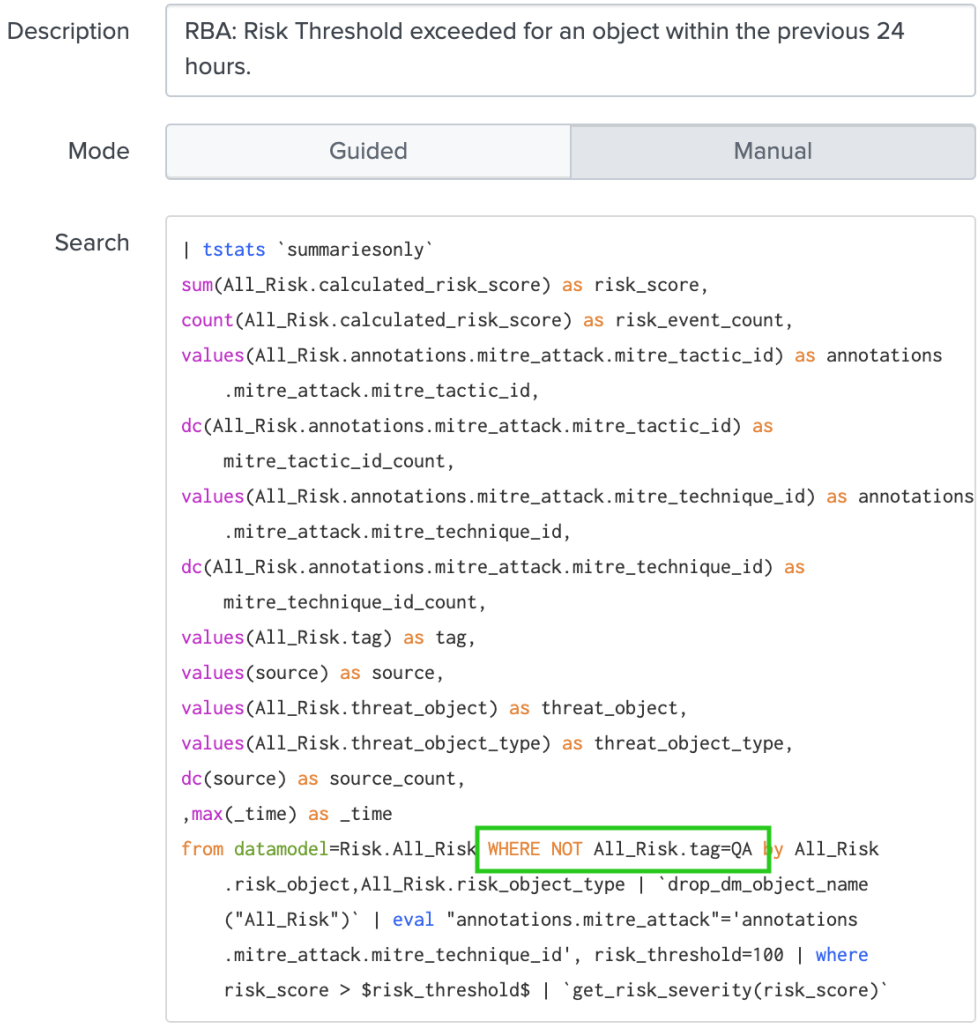


    b. Click the **New Event Type** button in the top right corner.
    c. Then, create an Event Type by entering information into these fields and then click **Save**.
        - Name: choose something like QA
        - Search string: `QA=1`
        - Tags: **QA**
        - Priority: **1** (Highest)
    d. The next step (which I always forget until I'm troubleshooting) is to set the Sharing value to Global in the Event types window.



    e. Click **Permissions** and check that it is set to **All Apps** and **Read** access for everyone.

3. Finish setting up QA mode by adding this exclusion into all the Risk Incident Rules. To do that, you add `WHERE NOT All_Risk.tag=QA` as shown in the image below.



Voila! You now have the awesome ability to run rules in QA mode. You can build out a week's worth of new Risk Rules in QA mode and let them run for a few days to generate risk events. Then, copy the Risk Incident Rule logic into a regular search and take out the exclusion to see a "before and after" of how many notables you would have if these were in production. This is a great way to see the effects of rule changes before they affect your carefully balanced risk ecology and ensure that you maintain a realistic amount of notables for your analysts as you create more content and close more gaps.

Extra Credit: Start Review and Response Processes

If you can get analysts or other SOC team members involved at this level to start reviewing how you respond to alerts, you'll be even better prepared for Level 3. Once you have some interesting Risk Notables surfacing, try to pull in an analyst to review alerts for an hour or two every week and get some of their feedback. Whether you use manual playbooks, automated workflows, or full-on case management tools, your response actions will probably need to adapt to the higher-fidelity alerts produced by RBA, and analysts are going to be your best source of feedback for what they need.

Extra Credit: Rising or Falling Risk Notable Deduplication

At about the nine-minute mark of an excellent .conf19 talk from Stuart McIntosh of Outpost Security, he mentions a really handy technique for dealing with Risk Notables that continue to fire as a risk object gains additional events and as earlier events start dropping off. In a production queue, this often means you're working an alert where most of the previous activity was considered benign, and it might not even be any truly new, unique events in the alert depending on how things are deduplicated or throttled. To solve this, he created a truth table that looks at the status of previous notables for that risk object whenever a new notable fires:

alert	previous notable	previous status	matched rules	matched score
yes	FALSE	FALSE	FALSE	FALSE
yes	TRUE	non-malicious	FALSE	FALSE
yes	TRUE	non-malicious	FALSE	TRUE
yes	TRUE	non-malicious	TRUE	FALSE
no	TRUE	non-malicious	TRUE	TRUE
yes	TRUE	malicous	FALSE	FALSE
yes	TRUE	malicous	FALSE	TRUE
yes	TRUE	malicous	TRUE	FALSE
no	TRUE	malicous	TRUE	TRUE

A truth table for new notables to reference old ones and decide whether we want to alert again or not.

He chose to not alert when the status was non-malicious, the rules matched, and the score matched, but you could also not alert when status is non-malicious, the rules matched, and the score doesn't match depending on your risk tolerance and how events bubble up in your RBA.

I've seen multiple customers solve this in their SOAR tools or with a similar approach, but one simple solution for this is to include any recent notables for risk objects in your Risk Incident Rule logic so analysts have that context available right away. For example, you could create a saved search running somewhat frequently which outputs a lookup of notables in the past seven days, then insert something like this into your RIRs:

```
| lookup Past7DayNotables.csv risk_object
 OUTPUT prev_time prev_status prev_sources
| eval prev_alerts = prev_time." - ".prev_status." - ".prev_sources
```

Then, make sure to configure `prev_alerts` to appear in Incident Review by adding them in **Configure → Incident Management → Incident Review Settings**. Now your analysts can get a brief overview of the previous alert, its contributing events, and if anything differs from their previous analysis, or could use `prev_sources` as a potential field for throttling.

Level 2 Checkpoint

Here's how you'll know if you're ready to move from Level 2 to Level 3 with RBA.

Regular Risk Rule Monitoring

The techniques presented in this chapter are not meant to be followed once and forgotten; they should be part of an ongoing strategy to derive value from your events. Early in Level 2, you'll spend a decent amount of time doing this, but as your risk ecology settles and you achieve an amount of stability, you will be able to spot disturbances or discrepancies. Additionally, you'll have done this enough times to be more familiar with what and where to investigate to bring balance back to your risk kingdom.

It's also worth mentioning that because you're now monitoring for more subtle changes in your environment, you may pick up on unexpected events. Monthly, quarterly or yearly IT maintenance may cause an explosion of risk, or maybe a new product is being onboarded into the environment which wasn't part of your initial tuning. Security should be seeing these things and knowing they're happening! Once you have a stable risk ecology, you'll spend less time performing ongoing tuning and these events will stick out like a sore thumb.

Interesting Risk Notables

After you've deployed a dozen or two rules, you should start seeing some interesting activity in your environment. Even if they're not always malicious events, I generally see customers come across privileged user activity, systems with abnormal behavior, or security misconfigurations. For many of them, it becomes a conversation starter with other teams, which helps flesh out the overall cybersecurity posture of their environment and position additional opportunities for improvement.

If this hasn't occurred, that doesn't necessarily mean anything is wrong. There is a possibility this may indicate that there are some types of detections, content gaps, or log sources you should prioritize to get a more diverse collection of events. It may be worth engaging your red team to run some tests with you and build content from the fingerprints of their activity in your available log sources, or try utilizing Splunk Attack Range to ensure you're picking up the right events for classic attacker activity.

Risk Notable Volume

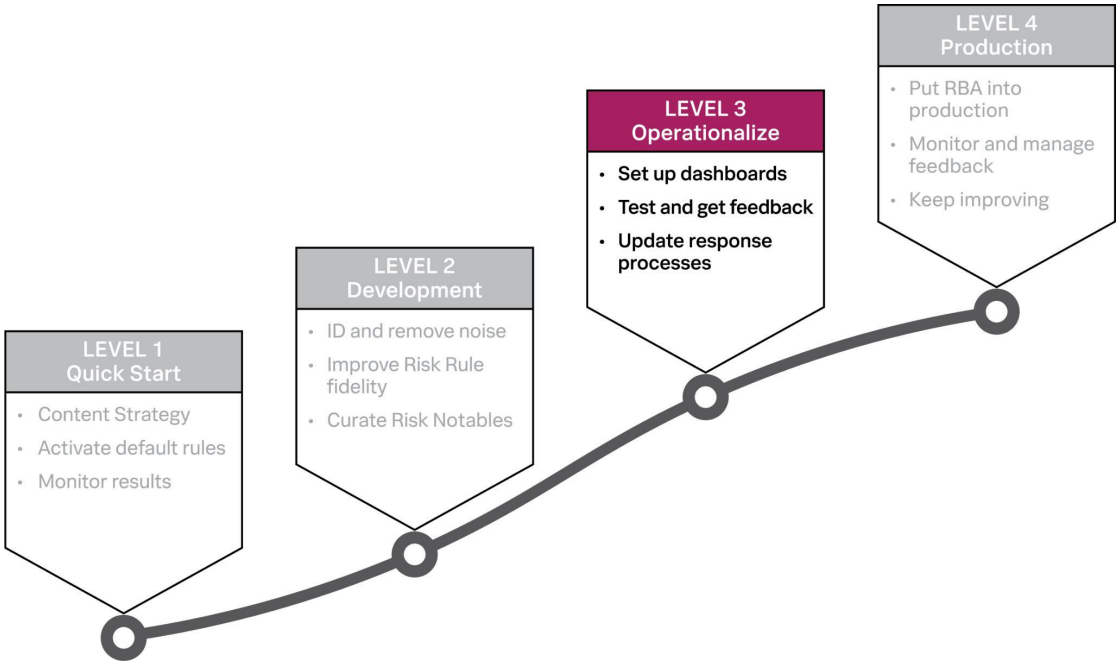
Part of the reason you're building out RBA is to reduce the amount of noisy alerts you're dealing with, right? Well, this actually isn't guaranteed, depending on the visibility of security detections you currently operate in production. If your previous strategy was "only respond to critical severity alerts from tools we don't bother to customize" or "whatever the MDR or MSSP teams say" then you may have limited volume in an artificial and potentially dangerous way. I think it's smart to keep in mind that a program which only responds to certain, always bad alerts is missing visibility into more complex attacks. It's also unlikely to catch novel threats your tools haven't developed alerts for, or you cannot realistically detect without integration into a framework like RBA.

Utilizing techniques provided in this chapter should result in a vast reduction of your initial Risk Notable volume, or help keep the amount reasonable in comparison to your traditional alerting pipeline. You want to make sure that whatever you do implement, the amount of time to deal with these alerts is less than what your analysts are currently spending to simply respond to an alert queue. The reason I love RBA is because of empathy for analysts that don't have time for training, projects to improve SOC operations, content maturation, or threat hunting… all the fun stuff!

You can expand the amount of time spent on these high-value activities by ensuring your Risk Notables are higher fidelity and actionable and by how you streamline the investigation and response process. We're about to get into it, so buckle up!

Chapter 6. Level 3: Operationalize

Welcome to Level 3, which is all about getting ready to move RBA into your production environment. There are three things to focus on in your preparation: testing and getting feedback from stakeholders, customizing dashboards, and updating your response processes.



Let's streamline processes to ensure a successful implementation!

Goals

- Engage relevant teams in your organization
- Streamline investigation dashboards and response process
- Ensure RBA quality is ready for production

Steps

- Devise continuous testing and feedback process
- Familiarize yourself with building dashboards
- Update response processes: playbooks, workflows, case management, etc.
- Implement in Splunk

Step 1: Continuous Testing and Feedback

First off, give yourself and your team a pat on the back for reaching Level 3! You are so close to getting RBA in production and that's really exciting. A lot of what I reference in this chapter will be familiar from the previous Levels, but in this one, we're focused solely on making sure things are fleshed out and working nicely before moving RBA into production. As you may have noticed, I've consistently mentioned how important and helpful it can be to bring in other teams where you can. As you get closer to using RBA in production, that kind of engagement is an absolute must!

Different teams have different needs, so whatever metrics you can provide to help them reach their goals will ensure that RBA makes a positive impact in your organization. Additionally, giving operational teams the opportunity to test-drive what you're building means you can implement changes from their feedback before they're forced to utilize a new process which may not have had their ideal workflow in mind. Obviously, you should be talking to team leads, but if they have people who will be responding to alerts, working queues, or meeting SLAs, then you have the opportunity to streamline their response within Splunk. With some SPL and dashboard magick, you can make their lives so much easier and you'll have RBA buddies for life.

You are also a customer of your own RBA development, so having dashboards with relevant metrics to guide your testing and implement your own feedback is just as important. As I mentioned in the checkpoint for Level 2, it's important to have a process for monitoring Risk Rules and Risk Notables. Tracking that week-to-week progress as your volume reduces and fidelity improves is not just an important metric for proving the value of the project to leadership, it also helps you guide and direct the development of your RBA program. Utilizing the `QA=1` method for Risk Rules and Risk Incident Rules is a great way to see the downstream effects of your development and implement feedback.

Step 2: Building Out Dashboards

To actualize the advice you've just read, I've designed this section as an introduction to dashboards because of how powerful and flexible they are. Even if you already know what they can do, I've included some fun tips and tricks that I want to make sure you know!

We've touched on the Splunk-standard Risk Analysis dashboard earlier in this guide, and it's a great way to start looking at where risk is building up or for basic risk object investigation. You can definitely use that dashboard during your entire RBA development process, but Splunk also offers amazing customization capabilities so that you have the opportunity to make each step of the process – content development, analysis, and reporting – uniquely streamlined for your needs.

One of my favorite parts about Splunk is what you're able to do with dashboards; you can find excellent examples from various apps on Splunkbase or build your own after messing around with the excellent app Splunk Dashboard Examples. Regardless, I highly recommend creating them to make your job – or anyone else's – much easier.

Dashboard Inspirations

Here are some ideas for dashboards, based on what I've seen customers build:

- **Risk Notable Analysis** for finding noise
- **Risk Notable Investigation** for streamlining analysis
- **Allowlisting/Tuning** for SOC analysts to modify lookups and reduce noise
- **Content Development** for tracking progress (or use the dashboards in the MITRE ATT&CK App or Splunk Security Essentials)
- **SOC Metrics** for comparing traditional alerting and RBA

Every customer I've talked to has a different environment and security situation, so you'll have to use your best judgment to figure out what will help RBA succeed in transforming your security approach.

Building a Risk Notable Analysis Dashboard

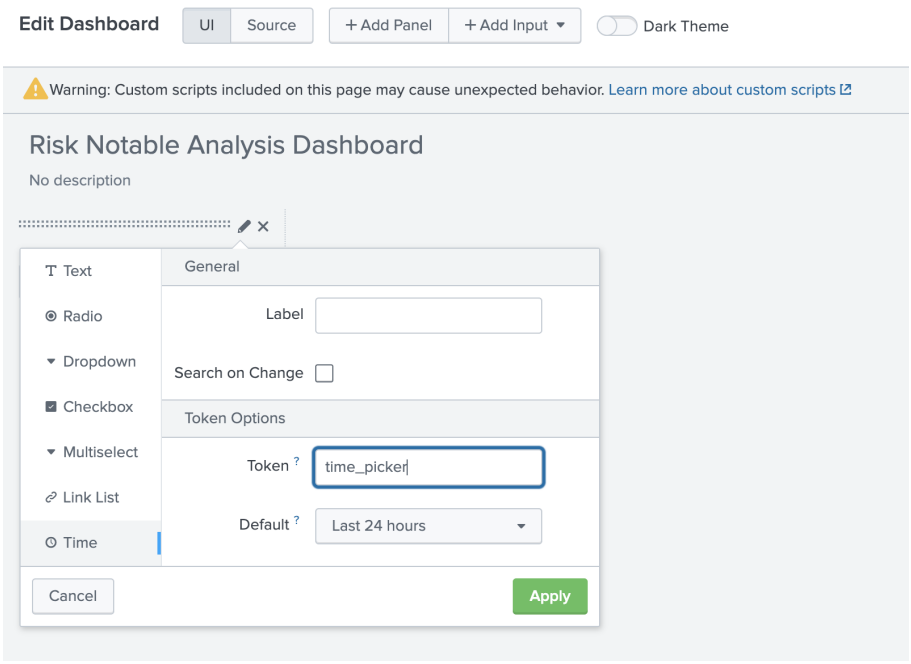
I'm going to walk you through building a Risk Notable Analysis dashboard in the following pages, but we'll cover techniques that can be helpful for any type of dashboard you build.

If you're not super familiar with building dashboards, don't worry! The GUI editor has evolved over the years and is super easy to use. If you watched my .conf20 talk "Streamlining Analysis of Security Stories with Risk-Based Alerting" you'll know how passionate I am about UX and interaction design (and you also got to see a **pretty freaking sweet** investigation dashboard). I just love when things I know I'll need for decision making or need to do next are within

grasp. (It's also the reason I love learning keyboard shortcuts when I start using a new application; CTRL+TAB, CTRL+T, and CTRL+W are my best friends when using a web browser.)

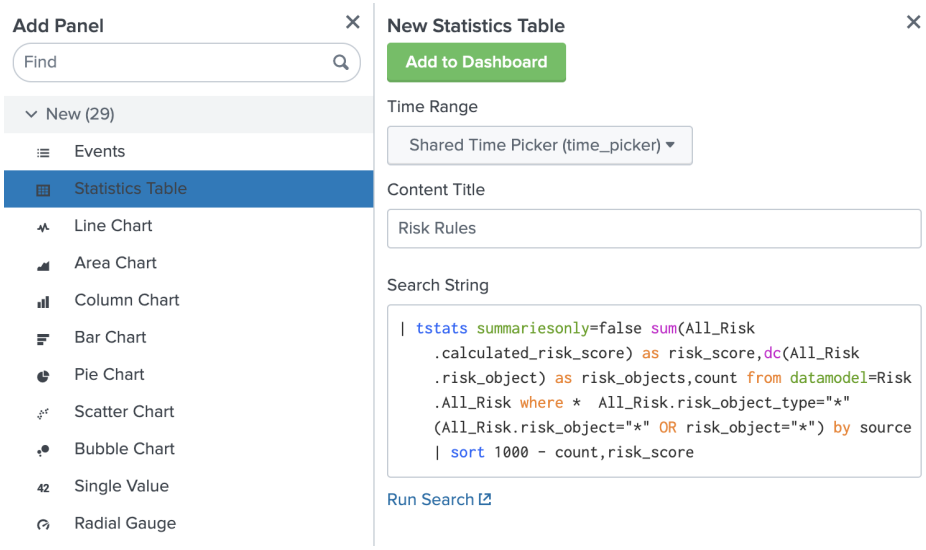
Let's build a dashboard with drilldowns utilizing some of the searches I mentioned in Level 2.

1. In the Dashboards panel, click the **Create New Dashboard** button on the upper right.
2. Give the new dashboard a name, perhaps something like *Risk Notable Analysis*.
3. Create a time picker and a submit button, then click the pencil edit button on the time picker to give it a token with a recognizable name:

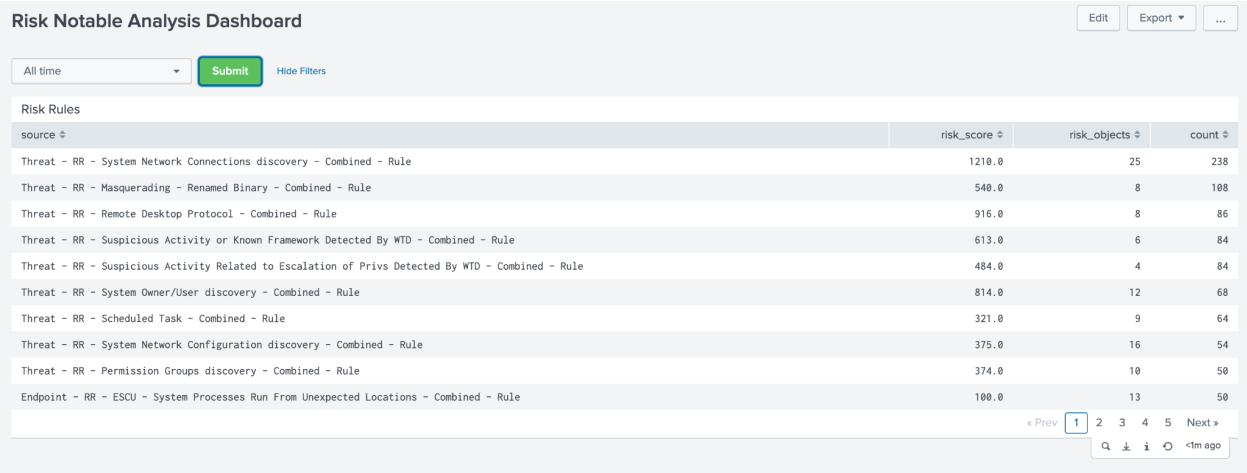


4. I like editing tokens and panels to give them helpful names as I build, but feel free to use your own. Let's use that first search from back in Level 2 and create a new statistics panel to show what searches are generating the most risk. To do that,
    - Click + Add Panel in the top left, then select **New** → **Statistics Table** in the column that appears.
    - Set the Time Range to **Shared Time Picker (`time_picker`)**.
    - Use **Risk Rules** in the Content Title.
    - Then, enter this code into the search string field:

```
| tstats summariesonly=false sum(All_Risk.calculated_risk_score)
   as risk_score,dc(All_Risk.risk_object)
   as risk_objects,count
 FROM datamodel=Risk.All_Risk
 WHERE * All_Risk.risk_object_type="*" (All_Risk.risk_object="*" OR
risk_object="*")
 BY source
| sort 1000 - count risk_score
```



5. Make sure you use the shared time picker so all the panels are on the same page, then click **Add to Dashboard** and check out how it looks:

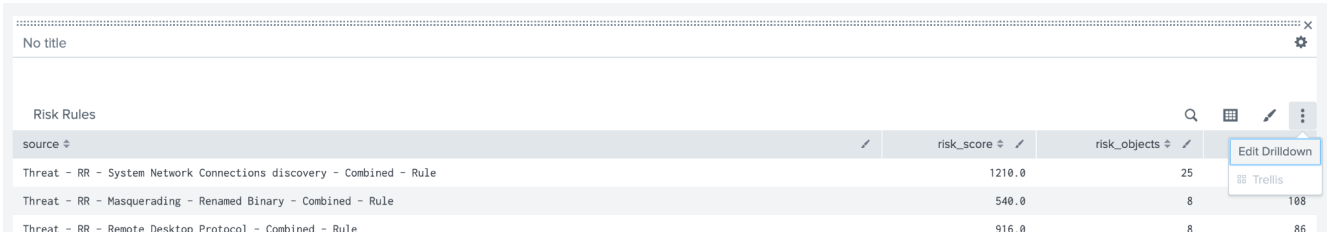


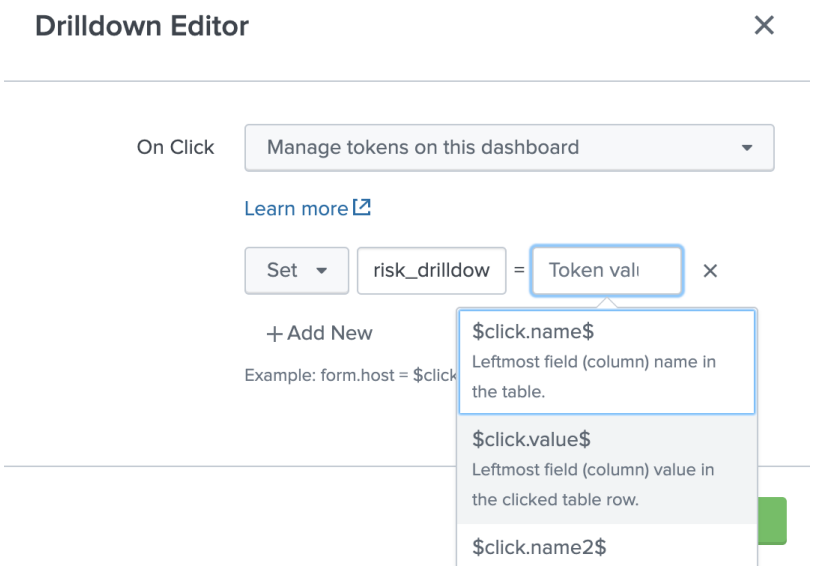
Well, that's a good start, but let's polish it up, shall we? Let's add in a drilldown to investigate the Risk Rule you want to look at, using a combination of the searches I mentioned immediately afterward in Level 2.

6. Create another statistics panel:
    - Set the Time Range to **Shared Time Picker (`time_picker`)**.
    - Use **`$risk_drilldown$`** as the Content Title.
    - Copy this SPL and replace the value for `source` with a token, like this:

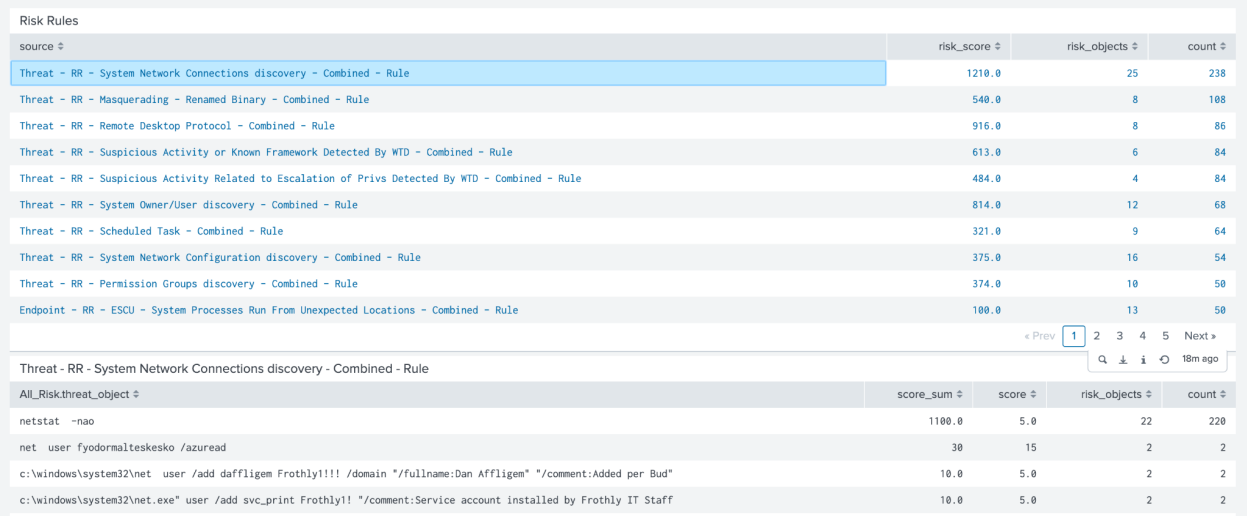
```
| tstats summariesonly=true
count dc(All_Risk.risk_object)
   as dc_objects dc(All_Risk.src)
   as dc_src dc(All_Risk.dest)
   as dc_dest dc(All_Risk.user)
   as dc_users dc(All_Risk.user_bunit)
   as dc_bunit sum(All_Risk.calculated_risk_score)
   as risk_sum values(All_Risk.calculated_risk_score)
   as risk_scores
 FROM datamodel=Risk.All_Risk
 WHERE source="$risk_drilldown$"
 BY All_Risk.threat_object,All_Risk.threat_object_type
| `drop_dm_object_name("All_Risk")`
| sort 1000 - risk_sum
```

7. Create a drilldown by clicking the three dots in the top right of the panel, and click **Edit Drilldown**:





This example uses `$click.value$` so no matter what field is clicked, it will use the search name in the first column to populate the `$risk_drilldown$` token. Check it out!



Risk Rules

source	risk_score	risk_objects	count
Threat - RR - System Network Connections discovery - Combined - Rule	1210.0	25	238
Threat - RR - Masquerading - Renamed Binary - Combined - Rule	540.0	8	108
Threat - RR - Remote Desktop Protocol - Combined - Rule	916.0	8	86
Threat - RR - Suspicious Activity or Known Framework Detected By WTD - Combined - Rule	613.0	6	84
Threat - RR - Suspicious Activity Related to Escalation of Privs Detected By WTD - Combined - Rule	484.0	4	84
Threat - RR - System Owner/User discovery - Combined - Rule	814.0	12	68
Threat - RR - Scheduled Task - Combined - Rule	321.0	9	64
Threat - RR - System Network Configuration discovery - Combined - Rule	375.0	16	54
Threat - RR - Permission Groups discovery - Combined - Rule	374.0	10	50
Endpoint - RR - ESCU - System Processes Run From Unexpected Locations - Combined - Rule	100.0	13	50

« Prev 1 2 3 4 5 Next »

18m ago

Threat - RR - System Network Connections discovery - Combined - Rule

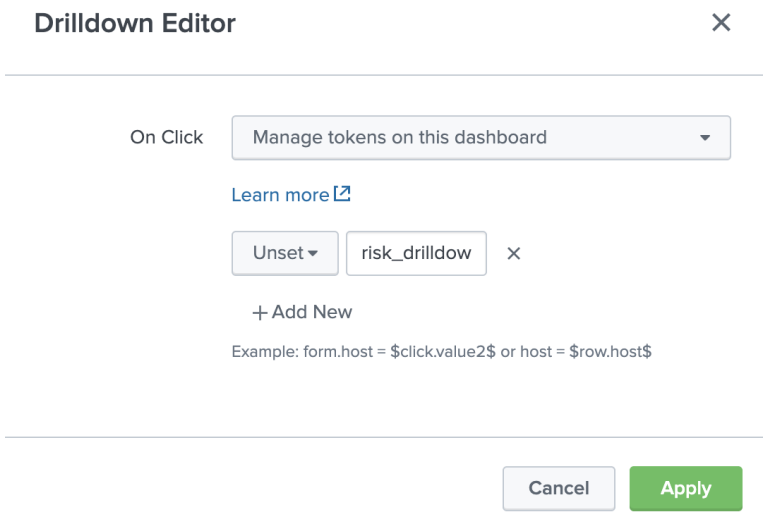
All_Risk.threat_object	score_sum	score	risk_objects	count
netstat -nao	1100.0	5.0	22	220
net user fyodormalteskesko /azuread	30	15	2	2
c:\windows\system32\net user /add daffligem Frothly1!!! /domain "/fullname:Dan Affligem" "/comment:Added per Bud"	10.0	5.0	2	2
c:\windows\system32\net.exe" user /add svc_print Frothly1! "/comment:Service account installed by Frothly IT Staff	10.0	5.0	2	2

And right away, you can see that **netstat -nao** is creating more risk by itself than all the other rules, so it might be worth figuring out how to tune it effectively. I've included an optional addition to this dashboard to dig into threat objects in Appendix C if you want to try it out for yourself. It gets a little deeper into Simple XML but enables some really interesting functionality.

Developing a Disappearing Dashboard

To keep the dashboard from getting cluttered while you investigate, you can add a drilldown so it disappears when you don't need it.

1. Let's **unset** the token when the user clicks something:



2. But, that doesn't actually make it disappear on its own. You need to dive into the source XML and add a depends statement. Enter edit mode and switch to source view:

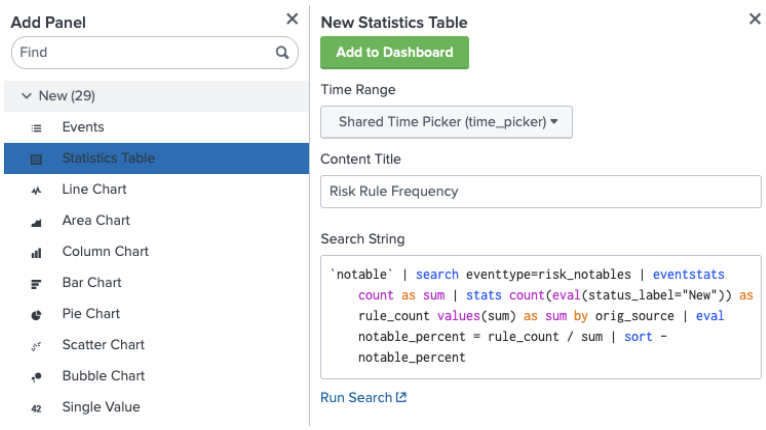
```
      </search>
      <option name="drilldown">cell</option>
      <drilldown>
        <set token="risk_drilldown">$click.value$</set>
      </drilldown>
    </table>
  </panel>
</row>
<row>
  <panel depends="$risk_drilldown$">
    <table>
      <title>$risk_drilldown$</title>
      <search>
        <query>| tstats sum(All_Risk.calculated_risk_score) as score_sum,values(All_Risk.calc
          source="$risk_drilldown$" by All_Risk.threat_object | sort - score_sum</query>
        <earliest>$time_picker.earliest$</earliest>
        <latest>$time_picker.latest$</latest>
```

Now the drilldown will be dismissed when you click any of the values in the new panel. Pretty cool.

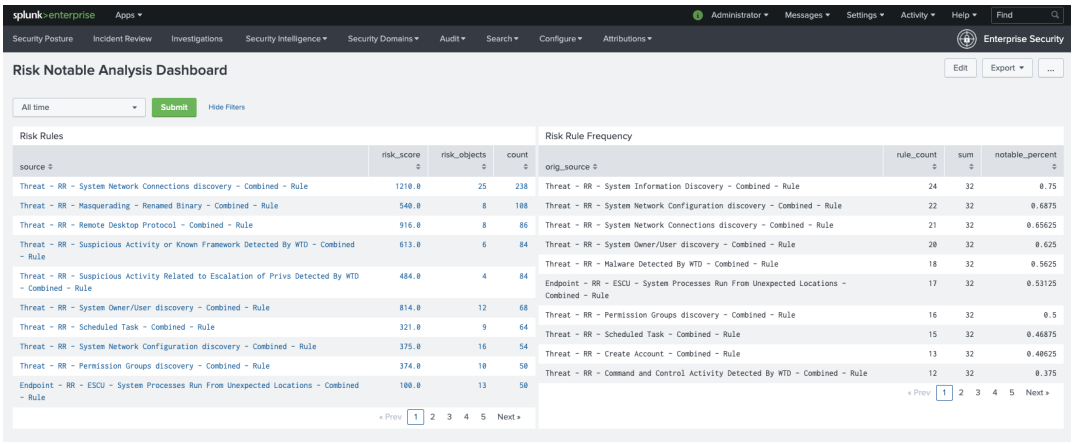
One more thing…

Before we wrap up, let's add one more useful search I mentioned in Level 1: how to identify what correlation searches are making too much noise in traditional alerts. Let's modify the search slightly to see how often each Risk Rule ends up in the Risk Notables:

```
`notable`
| search eventtype=risk_notables
| eventstats count as sum
| stats count(eval(status_label="New")) as rule_count values(sum) as sum
 BY orig_source
| eval notable_percent = rule_count / sum
| sort - notable_percent
```



I went ahead and moved it next to the top row, where it makes sense while you're tuning. Awesome!

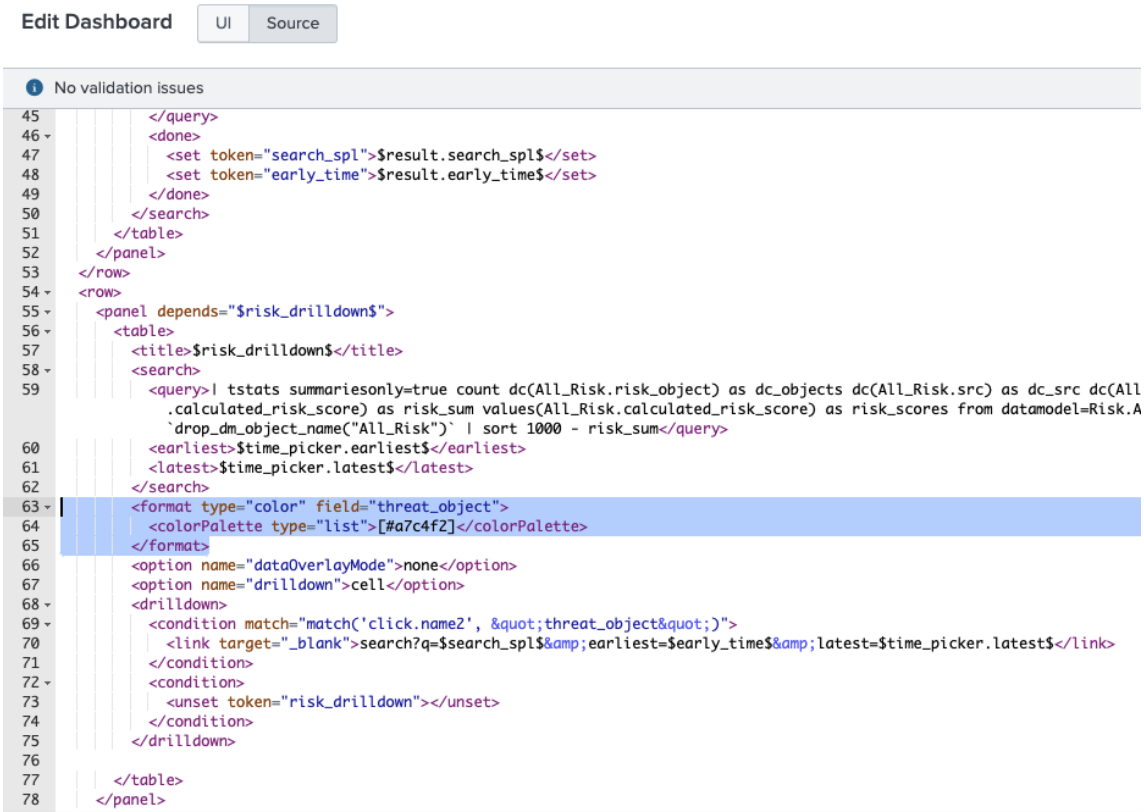


Color-coding your dashboards

One last thing before I let your brain roam free on how you're going to make sweet dashboards to gather metrics, testing, and feedback. There's a concept in the User Experience/Interaction Design world I love called "information scent" based on how animals hunt for food, and humans hunt for information. One of the ways we can indicate an information scent for a human to follow is by using color, so let's add a little extra Simple XML in the threat object panel to do just that.

There are all sorts of Simple XML formatting options for panels, but let's just add a color to the `threat_object` field to indicate that something will happen when you click it. You have to place this somewhere inside the `<table>` element, like this:

```
<format type="color" field="threat_object">
<colorPalette type="list">[#a7c4f2]</colorPalette>
</format>
```



Now check it out:

Threat - RR - System Network Connections discovery - Combined - Rule

threat_object	threat_object_type	count	dc_objects	dc_src	dc_dest	dc_users	dc_bunit	risk_sum	risk_scores
netstat -nao	command	220	22	0	21	1	0	1100.0	5.0
net user fyodormalteskesko /azuread	command	2	2	0	1	1	1	30	15
c:\windows\system32\net user /add daffligem Frothly1!!! /domain "/fullname:Dan Affligem" "/comment:Added per Bud"	command	2	2	0	1	1	0	10.0	5.0
c:\windows\system32\net.exe" user /add svc_print Frothly1! "/comment:Service account installed by Frothly IT Staff	command	2	2	0	1	1	0	10.0	5.0
net use	command	2	2	0	1	1	0	10.0	5.0
net user	command	2	2	0	1	1	0	10.0	5.0
net user /add Evil Account	command	2	2	0	1	1	0	10.0	5.0
net user /add rufus Gn@rly_Dud3	command	2	2	0	1	1	0	10.0	5.0
net user /domain	command	2	2	0	1	1	0	10.0	5.0
net user rufus	command	2	2	0	1	1	0	10.0	5.0

You might even notice I utilized blue to subconsciously align with how the web has used blue to indicate links since its inception. Now it's up to you to devise your own dashboards for making all sorts of jobs easier!

To help you out, I've uploaded the Simple XML for this dashboard to a GitHub repository in case you need it. I also uploaded some of the great dashboards by my RBA mentor, Jim Apger:

- Attack Matrix Risk - great metrics for before / after RBA and detection coverage
- Audit Attribution Analytics - useful visualizations for tuning, also a great summary table for monitoring correlation search results
- Risk Attributions - a ton of useful panels you might want in an investigation dashboard

Speaking of an investigation dashboard, I built one you're more than welcome to use with a lot of the handy features we discussed in this chapter (remember, try clicking anything blue; don't miss the previous notable count or threat object drilldowns). This walkthrough was basically a long-winded scheme for you to understand what's going on under the hood so you can mix and match from these resources to your heart's content. Give it a whirl and have fun!

Step 3: Designing Response Processes

RBA transforms how we detect, but we also need to keep in mind how someone responds to those detections. Hopefully you've listened to my gentle nudging throughout this guide to get other teams involved, because it's more necessary than ever as you approach replacing or supplementing your current security alert queue. A Risk Notable has to be investigated in a different way with a different mindset, and so you want to make sure when analysts work those alerts, they have a logical, repeatable process to follow. Or, when your SOAR platform receives those alerts, it provides necessary information for enrichment and potentially an automated response.

Most SOC teams have defined processes for responding to events, and these must be redesigned to work with the style of analysis required by RBA. Generally they seem to follow a pattern of:

1. Check whether the thing is bad:
    - Submit IP to URLHaus for analysis.
    - Submit file sample to VirusTotal for analysis.
    - <do more checking as needed>
2. Decision Point: Is the thing bad?
    - If yes, escalate to the Incident Response team: create a ticket and reach out to the on-duty team.
    - If no, leave comments describing what happened and resolve the alert with the status "Incident" if you created a ticket, or "Closed" if the alert was not malicious activity.

This is just an example; in real life, there are often many more steps and substeps, but the general idea of *if this, then that* is there… and we don't want to stray too far from this model for RBA. Score is a good start, but what else?

If/Then with Threat Objects

Now that we have multiple events contributing to the Risk Notable, you need to have some kind of modular process for analyzing each event and wrapping it up into how they fit together. One of my favorite response playbooks I've seen clever Splunk customers create uses threat object types to decide the next steps: one playbook for when a `file_hash` is a threat object, another for `ip_address`, another for `command_line`, and so on, giving each artifact its own playbook. Additionally, what I love about this approach is how this can be married with SOAR automation to enrich the threat object and potentially even respond or close the alert based on its results. The RBA Playbook Pack by Kelby Shelton is a great way to get this kind of approach going in Splunk SOAR.

If/Then with Risk Rules

I want to reiterate how you can provide useful context to include as part of the response process. I recently met with a customer who had this incredible idea; I realized the value of their clever count / sum SPL while writing this guide. For each Risk Rule in their Risk Notable, they also included how many times those risk events bubbled up into a Risk Notable, as well as whether it was closed out or responded to. The analyst looking at the list of events can easily see differences:

- A rule fires regularly and associated Risk Notables are closed 95% of the time.
- A rule fires infrequently but is often closed out.
- An infrequent event occurred but when it does, it is often involved in an actual incident or required some response.

This differentiation helped their analysts use that information to determine which events might be the most relevant threat objects in an alert. Combined with the risk score, this gave analysts a headstart on investigating the most egregious or potentially dangerous activity first and framing their analysis accordingly.

If/Then with Risk Object Attributes

Lastly, we have information about the risk object. One of my favorite things to know about a user is their business unit, because this helps me frame the type of activity they are likely to do. Even when I receive an alert for a system or device, I also like knowing the business unit of the system owner for the same reason. Incorporating step-by-step processes for framing the investigation of an alert with the context around who or what you're investigating helps inform the analysis of each RIsk Rule and threat object going forward. This is why fleshing out your Asset & Identity framework can be so powerful!

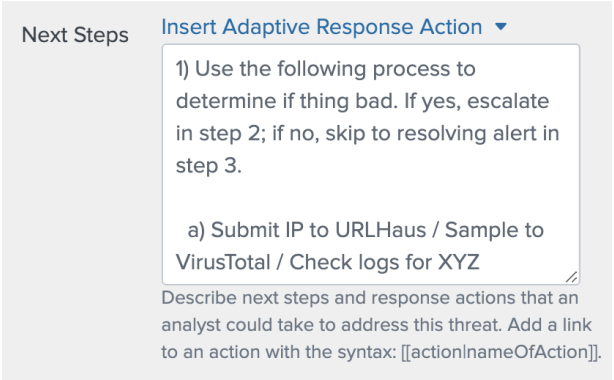
Every environment is different so I don't want to get too prescriptive here, but hopefully those examples give ideas around how people work with this new kind of alerting. If you and the team responsible for your alerts work together, you'll be able to craft something that makes the response to your detections even more effective and appealing to the people working those alerts in real time.

Implementing in Splunk ES

Now that you have a process, let's briefly review how you can implement it into Splunk. Not everyone utilizes Enterprise Security as their ticketing and response tool, but whether or not you do, what we're doing here is worthwhile to think about, even in the context of an external platform.

Adding Process Steps to Risk Notables

First, you want to include your step-by-step process into the Next Steps section of the Risk Notables in the Incident Review panel. You edit this in the **Notable Adaptive Response action** of the **Risk Incident Rule** correlation searches so the process steps are immediately available to the analyst investigating the alert.



Modifying the Incident Review Dashboard.

Next, I like looking at the fields Incident Review, and I have thoughts. I lean toward a minimalist style that provides very broad strokes with the most important information when the initial alert arrives, then I curate detail into my investigation dashboards. Other folks like having a big list of fields with all of the juicy info to review as the alert arrives into Incident Review, before they dive into dashboards. I'll just briefly touch on configuring fields in Incident Review, then you can decide what works best for your team… with their input, of course!

The default settings for RBA with ES enrich the various MITRE ATT&CK techniques in the event with their corresponding tactic. Tactics may appear multiple times because of the enrichment on the techniques listed underneath.

Description:

Risk Threshold Exceeded for an object over a 24 hour period

Additional Fields	Value	Action
Mitre Tactic	discovery	
	lateral-movement	
	discovery	
	defense-evasion	
	discovery	
	execution	
	persistence	
	privilege-escalation	
	execution	
	persistence	
	privilege-escalation	
	discovery	
	discovery	
	discovery	
	discovery	
Mitre Technique	System Network Configuration Discovery	
	Remote Desktop Protocol	

This does offer the advantage of indicating there is overlapping activity in tactics (in this example, the attacker triggered lots of events in **discovery**), but I would personally decide to offer a little less detail here, or organize it differently. For example, if I edited the SPL of my Risk Incident Rule, I could make a new field at the end like this:

```
| eval mitre_tactics = 'annotations.mitre_attack.mitre_tactic_id'
| lookup mitre_enterprise_list.csv mitre_id AS mitre_tactics
 OUTPUT mitre_tactic AS mitre_tactics
```

With a simple lookup matching the MITRE ATT&CK TA#### fields to their human readable names.

Then, in **Configure** → **Incident Management** → **Incident Review Settings**, I can make these changes (and remember to save them!):

- Add the new `mitre_tactics` field
- Remove the `annotations.mitre_attack.mitre_tactic` field

Now my alerts would look like this, where we only show the MITRE ATT&CK tactics once, which I prefer, personally.

Additional Fields	Value	Action
Mitre Technique	System Network Configuration Discovery	
	Remote Desktop Protocol	
	System Owner/User Discovery	
	Masquerading	
	System Network Connections Discovery	
	Scheduled Task/Job	
	Scheduled Task	
	Process Discovery	
	Permission Groups Discovery	
	System Information Discovery	
	Domain Account	
Mitre Tactic Count	7	
ATT&CK Tactics	Execution	
	Persistence	
	Privilege Escalation	
	Defense Evasion	
	Discovery	
	Lateral Movement	
	Collection	
Mitre Technique Count	16	

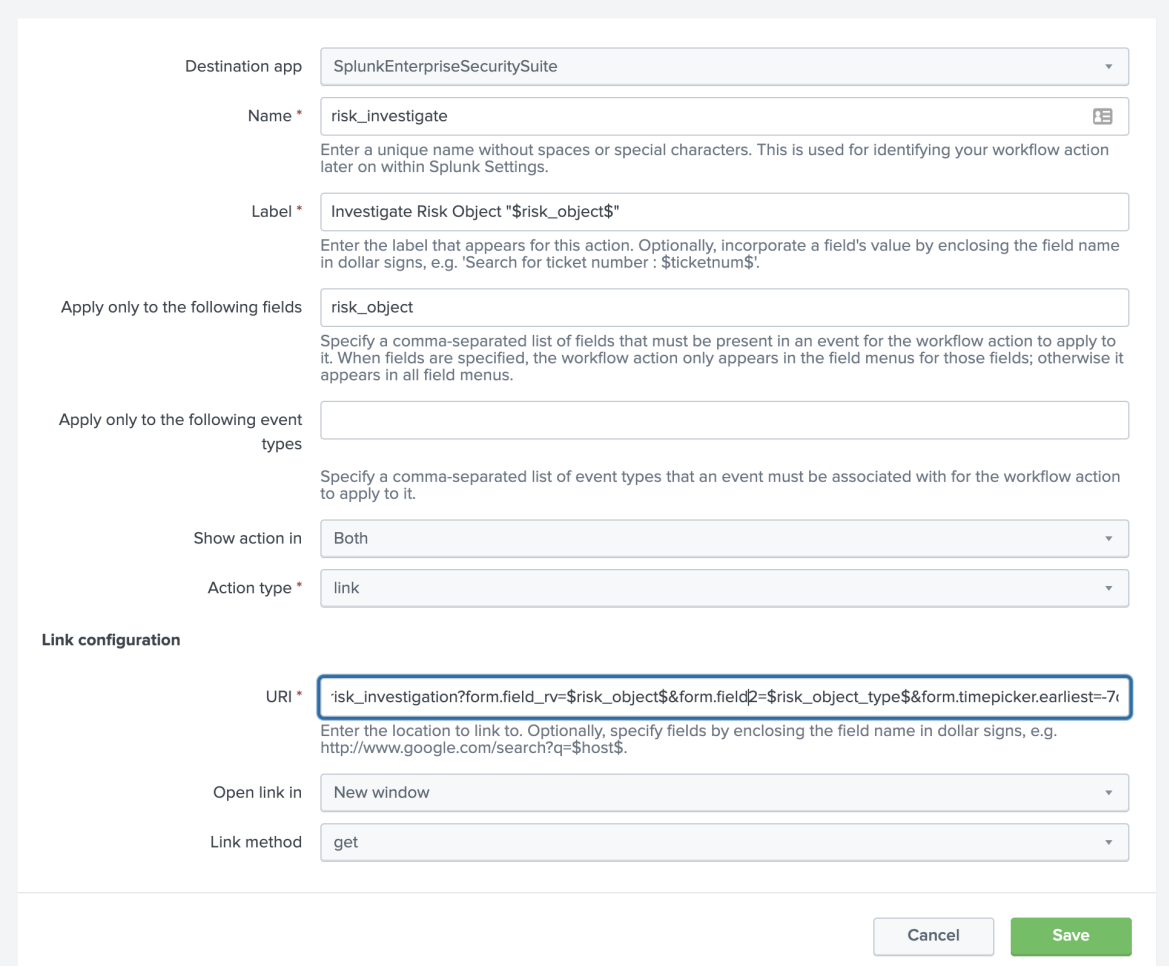
I might even take out the list of tactics and only have techniques, or replace them both with a list of the Risk Rules that fired instead. It's really up to you and your team and the best way to implement your workflow.

Linking to Workflow Actions

Speaking of workflows, we can create links to pivot from the Incident Review panel with workflow actions. These will appear within the dropdown arrow next to each of these fields, allowing you to include information from the event and pass them into a URI. We can use this to pivot directly into an investigation dashboard or even an external resource!

Let's say you created my investigation dashboard in your environment and you'd like to click a workflow action to open a new tab with field values from the alert prepopulated.

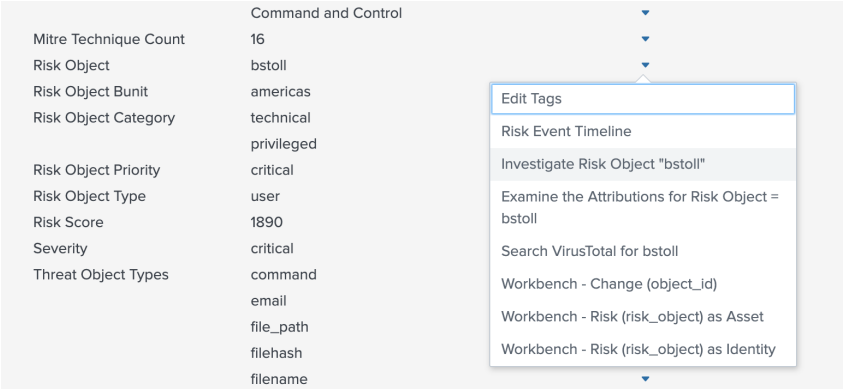
Go to **Settings → Fields → Add New Workflow Action** and you'll see this panel:



For the URI, I crafted this:

`https://my.splunk.instance/en-US/app/SplunkEnterpriseSecuritySuite/risk_investigation?`**`form.field_rv=$risk_object$&form.field2=$risk_object_type$`**`&form.timepicker.earliest=-7d&form.timepicker.latest=now&form.search_filter=*`

This will hit the URI of my dashboard `risk_investigation`, then the trailing question mark enables me to set tokens within the dashboard separated by ampersands for when I arrive. Now I can click the arrow next to the risk object field and then click my new workflow action "**Investigate Risk Object**" to pivot into my dashboard and begin investigating. Nice!



I'll leave it up to you to decide how to configure this aspect of your RBA implementation, or how to set up notables to play nicely with your downstream ticketing system. Just keep in mind how much flexibility you have with SPL to collect and transform information before you pass it along, so you have exactly what your team needs to effectively respond with their fancy new process.

Level 3 Checkpoint

After all the hard work you've put in on Level 2 and Level 3 tasks, you are SO close to going live with RBA. At this point, teamwork becomes even more important, right up there with developing and testing. Regular meetings with your team, reviewing playbook readiness, and selecting useful metrics are all good things before you move to Level 4.

Weekly Check-Ins

In Level 2, I talked about having regular sessions to dig into your Risk Index, monitor the amount of risk generated by Risk Rules, and see how many Risk Notables are generated. By now, you should be much more familiar with the ebbs and flows of risk in your environment and (if you haven't already) it's time to share your wisdom and guidance with the teams you'll be supporting. It's super helpful to have analysts pitch in an hour or two per week to investigate Risk Notables and get their feedback. At this point we are very close to putting RBA in production and I would not say it's helpful, it's necessary. There are so many things they can shed light on:

- What generates too much risk
- Any repetitive actions they have to perform
- Information they wish they had in Risk Rules or Risk Notables
- Ideas for new Risk Rules or Risk Notables

You might not be able to incorporate all of their feedback right away, but be sure to write it down for later improvement efforts. Getting people involved, asking for their expertise, and implementing their feedback demonstrates you care about the value they can provide. When you show cybersecurity folks some really cool cybersecurity things and then ask for their input and meaningfully respond to that feedback, they can't help but become excited about RBA and help you in all sorts of ways you can't predict. RBA is a team effort, so build your team!

Response Playbook

Now that you've enlisted the help of your response team and they've crafted a response playbook with their leadership, the big question remains: does it work? Make sure you get a week or two of additional feedback from analysts utilizing the playbook and ensure the results are satisfying business needs and requirements. Keep in mind where you can add context to events or craft a dashboard to provide additional efficiency to any step in an existing process.

RBA Metrics and Milestones

You've spent all of this time and effort to make an incredible RBA framework, so make sure you're telling the story that shows off how valuable it is! There are all sorts of interesting metrics that can support your RBA story:

- Number of alerts per day. Depending on how narrowly defined and limited the scope of your alerting was before RBA, the more meaningful metric may be something like "Number of events being observed, integrated, and alerted upon intelligently with RBA."
- Average time to triage an alert
- Percentage of alerts which result in no meaningful action

- Expected number of hours saved with this many alerts + time to triage
- Number of correlation searches on each security data source
- Percentage of MITRE ATT&CK tactics and techniques covered
- Time spent to create and curate detections

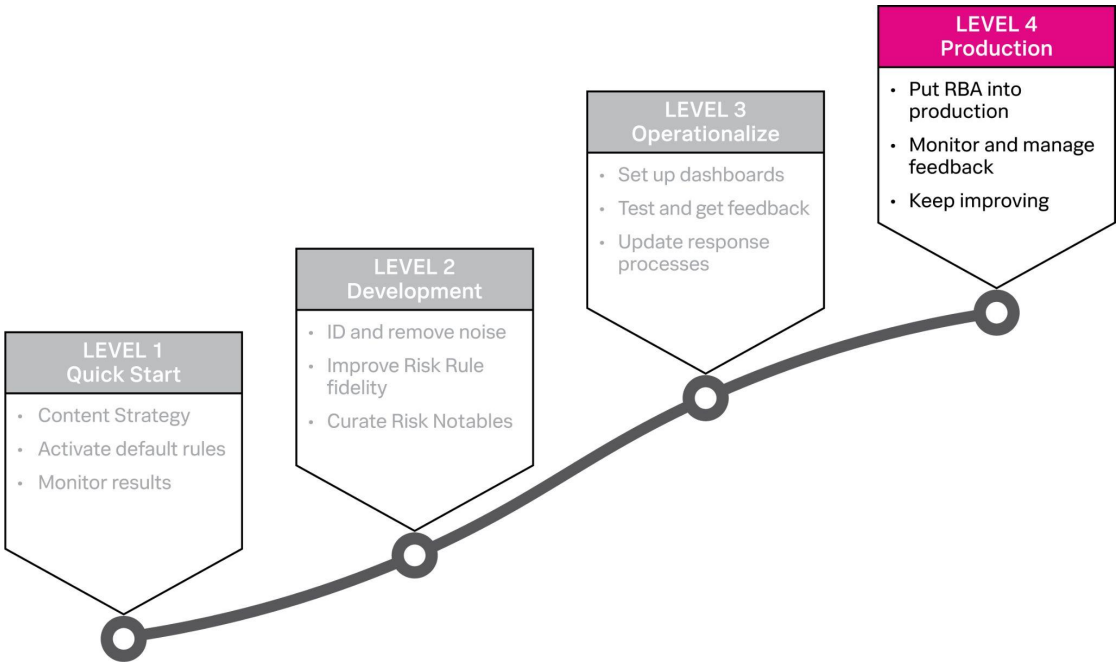
On top of pure quantitative metrics, there may be significant qualitative milestones:

- Additional value out of engagements with red team
- Able to craft and validate new detections with adversary simulation tools
- More communication and collaboration between teams
- New capabilities for threat hunting
- Bright new future of content development

That's just a few ideas. I'm so excited to hear about your success and please share with us on the community Slack!

Chapter 7. Level 4: Production

Once you've operationalized RBA, you're ready to move into your production environment. It's time for RBA to shine in the real world!



We're at the finish line; let's check a few boxes before we finally push RBA into production in Level Four!

Goals

- Start using RBA in production
- Be prepared to tune and improvise as needed after Go-Live
- Planning the future

Steps

1. Go-Live
2. Monitor and manage feedback
3. Engage additional security teams
4. Keep improving

Step 1: Go-Live

CONGRATULATIONS!

You've put in a ton of work to get here, so let me tell you the work will be so much easier from here on out. Now you get to witness the transformational power of RBA ripple through your security organization. You've established the foundation to build a flexible and adaptive cybersecurity program with an infinite security maturity ceiling. We are so close to an exciting future for your security program but now it's time for the true stress test: Go-Live.

Moving RBA from development to production is not a decision made by a single person. You'll need agreement from all of your stakeholders that RBA is truly ready for prime time: your SOC team, security analysts, red/blue teams, and so on. The metrics, feedback, and testing you've worked on in Level 3 will be an integral part of ensuring that RBA is truly ready to go.

Thinking about the day you Go-Live with RBA can be stressful, but by setting expectations and preparing you can make things a whole lot smoother. Your people should know:

- The basics of how events bubble up from risk events into a Risk Notable
- What RBA-generated alerts will look like, especially risk objects, risk messages, and threat objects
- Helpful context added by RBA in events or notables
- The response process for alerts
- Actual hands-on-keyboard steps to execute the response process and investigate Risk Notables
- How to provide feedback on all things RBA, including tuning and suggestions
- Who will be on-call for questions, troubleshooting, and unforeseen emergencies. If your security team runs 24/7, make sure you have someone on-call 24/7 for RBA questions or problems as well.

It may require more than a few meetings with the whole team to ensure everyone is on the same page, but solid preparation will save everyone from a lot of frustration when RBA moves into production. For the first week, I recommend having a check-in with analysts halfway through the day and / or at the end of the day to see how things are going.

You'll personally need to prepare as well. You should:

- Set aside the entire Go-Live day to make sure things go smoothly. Watch how your analysts respond and be prepared to make some tweaks to ease issues.
- Have a sheet handy with the traditional correlation searches you're disabling that have Risk Rule counterparts and documentation of what you're doing.
- Have a backup plan for reverting back to traditional alerting in case everything everywhere all at once goes wrong.
- Get a good night's sleep.

With all the preparation we've done in this guide, I'm confident you'll have a smooth transition and can approach Go-Live day with no fear!

Step 2: Monitor and Manage Feedback

After you introduce RBA into your production environment, keep having regular meetings with the same people who worked on the development and Go-Live planning and implementation. Ask how it's going, listen to what they're saying (and not saying), and work to address issues earlier rather than ignore them until they are massive problems.

If your team is already using metrics like Mean Time to Detect (MTTD) and Mean Time to Response (MTTR), you may need to explain how those metrics change when using RBA. For example, MTTD may increase because you're catching a series of events over time and only alerting when they hit a threshold rather than detecting and alerting on every single low-level event. Your MTTR may also change if analysts are spending more time analyzing complicated, high-fidelity threats.

Step 3: Engage Additional Security Teams

After you've collected all of these great metrics and had a (hopefully) smooth transition to RBA production, it's time to spread the RBA gospel. I generally get three responses when I introduce and explain RBA:

1. Holy moly! Why did nobody show me this earlier?
2. Oh, I've been doing something like this… but I didn't know you could do that!

3. Oh, I've been doing something like this… and mine also does *this*!

All of which are great in their own ways. What I love about RBA is how it can enable all sorts of teams to get value out of their data sources. Oftentimes, security is seen as the department which tells other teams "we can't do that" or "you can't do that", but with RBA you can say, "we can do this together!" By explaining how the system works and relying on their expertise, you can help devise what they would be able to do with the data sources they have. Then you can craft a small subset of Risk Rules and Risk Notables just for their team to help solve specific problems, generate an intelligent report, or give them an actually meaningful checkmark for their security audit.

As I mentioned in Chapter 1 of this guide, some of the most relevant and exciting applications for RBA are machine learning, insider risk, and fraud. This guide was designed with malicious compromise detection and response in mind, but you may have been following along in one of those other contexts. Regardless, now that you have RBA functioning in your environment, reach out to those other teams and show what you're able to do! Now that you've built out the foundation, it will be much easier to integrate additional modules or capabilities into RBA.

Step 4: Keep Improving

One of the biggest challenges with RBA management is convincing your team to keep tweaking your risk ecology to keep maximizing alert fidelity and minimizing noisy, low-level alerts. RBA may initially work so well that people think there's no more to be done but, as software developers already know, there's always more you can do to improve!

We've talked about different ways to use RBA throughout the guide, so if you skipped past an Extra Credit section then go back and see if you can make it happen. For gosh sakes, if you haven't used this for purple team exercises with red and blue teams, then get on it! Not only is it good for improving RBA, it's going to take your entire cybersecurity effort to the next level of maturity.

Level 4 Checkpoint

You made it! You're officially part of the "I Developed Risk Based Alerting and Didn't Even Get a T-Shirt" community. You're truly an RBA nerd now!

Throw a party

I'm serious. Celebrate your success. You deserve it! You and your team have achieved so much by building RBA together, and it's going to be the beginning of a new chapter. A lot of us are remote nowadays, but it's important to celebrate achievements with others to reflect on where we've been and where we're going. Whether it's taking the day off or having a zoom party, whatever works for your team to celebrate, do that.

Plug into the World of RBA

I definitely want to hear about your experience, so make sure to tell us how things went – or ask for help anytime – in the Outpost RBA Slack. I'm also constantly amazed by the unique ways people are utilizing or customizing RBA, so go talk about it and spread the gospel. RBA is consistently one of the most popular topics at Splunk's annual .conf, so submit a talk and share your solutions with others. I bet other conferences or vendors would be interested in how you took your security tooling to the next level with your specific RBA-based approach!

As you can probably tell, I'm so excited for the future of Risk Based Alerting and I truly thank you for reading all the way to the end.

Appendix A: Level Up Resources

One of the best things about Splunk is that core Splunk Processing Language (SPL) allows you to do so much with your data. I love that rather than submitting an idea to an idea portal and hoping other customers want the same thing fixed, then waiting for months or years before seeing any development happen, I could likely figure out how to make it happen within the Splunk sandbox.

On the other hand, the potential and power can be difficult to realize if you aren't able to connect the dots between what you know you know, what you know you don't know, and what you don't know you don't know. So I'm offering a few methods for finding what you need to learn as well as some essentials for Splunk in general, but especially what we're doing with security.

Learning How to Learn

I would be remiss to ignore that not every prescriptive method for learning how to do something effectively – like becoming better at Splunk – is going to work for every person. There are general themes supported by scientific research into memory and learning that likely apply to most people utilizing the human brain as their operating system, so I like to recommend my favorite resources to help you get a handle on learning how to learn:

- The Process of Mastering a Skill - article by Azeria (@Fox0x01)
- Learning How to Learn - course by Barbara Oakley and Dr. Terrence Sejnowksi
- How to Remember What You Learn - short video which distills insights from the above course

Splunk Resources

RBA Community

The wonderful folks at Outpost Security developed a Slack channel for people to chat about all things RBA way back in 2018, and it's been a great place for folks to ask questions, trade best practices, and connect with people at all stages of the journey. Some lovely folks from Splunk joined forces to organize monthly meetups and office hours and it's been a great way for people to stay connected and get regular feedback. There's also a community curated list of resources for RBA, which points to a ton of great resources. I'll see you there!

Hunting with Splunk Blogs

This links to a **ton** of great articles about how to do security with Splunk. I got a decent amount of my chops as an analyst and detection engineer thanks to these walk-throughs. To this day, these articles remain a great resource for leveraging Splunk for Security. On top of helping you craft some great security content, it will likely help you move up the leaderboard for the next Boss of the SOC! If you haven't been to the BOTS portal, it has a bunch of free modules for getting that hands on keyboard familiarity with a bunch of concepts mentioned in the blogs.

Product Documentation

As much as a lot of us like to just dive into a brick wall until we break through, you don't always have to. The Splunk docs contain a bunch of helpful information, but finding what you need can often be the challenge. Try to narrow down your request to a word or two of the feature you're trying to use, or even utilize a search engine with "*site:docs.splunk.com the thing you're figuring out*".

Lantern

Splunk Lantern contains a ton of helpful walkthroughs from Splunk experts on various aspects of the Splunk platform. I especially love the Security Use Cases section, and I've learned so much from various Tech Talk deep dives. I recommend searching around to find topics you're interested in, especially if you don't feel as confident as you would like to be.

There is also an excellent short resource for the overview of Implementing Risk Based Alerting which is a great primer to some of the concepts we dive into in this Guide.

Splunk Community

Us engineering types are very familiar with consulting Stack Overflow as part of the problem solving process, and Splunk Community is where you'll find so many answers to people in the same shoes as you. What's great is that even if you don't find an exact answer to a search, when people offer helpful SPL to solve a problem you'll often see something tangential you hadn't thought about that may be the key component you're missing! Finding that helpful answer may be the stickler, but as I recommended searching the docs, you might also want to try using your favorite search engine with "*site:community.splunk.com*" and your query, which sometimes works better on longer search phrases.

Splunk .Conf Presentations

I've sprinkled some of my favorite RBA talks throughout this guide and have all of them listed in Appendix D, but there are tons of great presentations out there on other topics and general Splunk usage. You can use the .conf Archive Search App to check what's available in previous years beyond the website. David Veuve's Security Ninjutsu talks opened my eyes to the possibilities of security with Splunk when I was just starting out, but there are so many more that have helped me learn the intricacies of leveraging my data with SPL. You'll find your own unique journey as you learn along the way!

Favorite SPL Commands

You definitely want to make sure you're pretty comfortable with general SPL like search and stats, but the flexible power of the following commands may not be immediately apparent:

Rex - Field creation, extraction, or standardization. Can extract new fields, or mode=sed for find + replace. If you're not familiar with regular expressions, I highly recommend this introduction.

Erex - An automated rex command; you can see the regex it generates by clicking the Job dropdown to the bottom right of the search panel.

Eval - Slice data, perform inline calculations, and juggle disparate data into more useful data.

Foreach - You programmer types want your for loops? We've got your for loops. This can be hard to grasp initially, but is so powerful when combined with regex skills.

Lookup / Inputlookup / Outputlookup - Storing things in a lookup (that I could build a dashboard for people to easily add to) is my favorite. Or using a lookup to perhaps store (with all the details) an alert that has fired before and bring useful metadata to new alerts.

Autoregress - For if this, then that alerts. Basically you can sort by user/src with `_time`, then check if the previous event is related and only alert/detect `| where user = prev_user`. This is one of my favorite commands to add risk with sequences of zero risk RBA events that may seem difficult to solve otherwise.

Makeresults - Sometimes you just need some sample data in a certain format to make sure your logic is working correctly, and makeresults is the friend you've always wanted. The documentation has a few excellent examples to help you get started.

Appendix B: Risk Incident Rule Ideas

We've seen folks come up with all sorts of interesting ways to alert, especially in unique RBA situations that aren't necessarily security related. I like to think about how cool it would be to have Risk Based Alerting based on Risk Factors of my ancestors' medical history and risk events generated from regular health tests! The other thing to keep in mind is setting different thresholds over different time periods, and you could keep these separate from your standard alerting so it's more of a threat hunting queue.

Here are some sample ideas you could play with:

- Risk Events from Multiple Sourcetypes over 24 Hours (one of my favorites)
- Anomalous Risk Score from Peers by Business Unit / Asset Category (> 2 stdDev from peers)
- Anomalous Score Trend for Role / Category (percentage increase)
- Anomalous Score Trend for Threat Object
- First Time Seen Threat Object Observed by Few Risk Objects (6 months)

I've also seen folks create custom frameworks which look for specific chains of events, or developing a "punch-card" (utilizing the Punch Card Visualization) to fingerprint interesting activity patterns and basic clustering with the Splunk Machine Learning Toolkit to rapidly identify bots or fraudulent activity. Please share your own creations or ask what people are up to in the Outpost Slack and I'll see you there!

Appendix C: Advanced Dashboarding

In the example dashboard from Level 3 (bonus points if you knew Dashboarding was in Level 3!), we encountered the pesky threat object *netstat -nao* which was causing way more risk than it should. If you were tuning, your next step would be to investigate why. How about setting up the dashboard to drill down into that pesky *netstat -nao* threat object we saw in my example and find out why it's so common when you click it?

To do this, you're going to dive into Simple XML code because I really want you to see some of the things it can do. If you're not familiar with XML, it's a series of nested `<commands>` with `"field=logic"` and `$variables$` quite similar to HTML. Splunk made a version called Simple XML to power our dashboards. You'll be using a combination of `<commands>` and `$tokens$` to juggle useful things with what the user clicks to make that capability operate smoothly.

Let's break down what we're going to do here, because it's going to seem a bit overwhelming at first. Don't be daunted, we'll work through it step by step! Go ahead and switch to the source code view and follow along with what we just did *mostly* with the UI in Level 3. The table below shows the actions and the Simple XML commands that actually do the heavy lifting to show the drilldown panel:

Action	Simple XML Code
When a user clicks the Risk Rule,	`$click.name$`
Set a token with that search's name…	`<set>` `$risk_drilldown$`
Then launch a previously hidden threat object panel just for that search using what was stored in `$risk_drilldown$`	`<drilldown>` `<panel depends="$risk_drilldown$">`

Now we're going to create the Simple XML code snippet that will automatically open a new tab with the correlation search generating these threat objects (and you can investigate that pesky netstat -nao, for example):

Action	Simple XML Code
Run a search for a panel we won't even see…	`<panel depends="$doesnt_exist$">`

To set new tokens with correlation search SPL…	`<set>`
	`$result.search_spl$`
	`$result.early_time$`

After the search is finished.	`<done>`

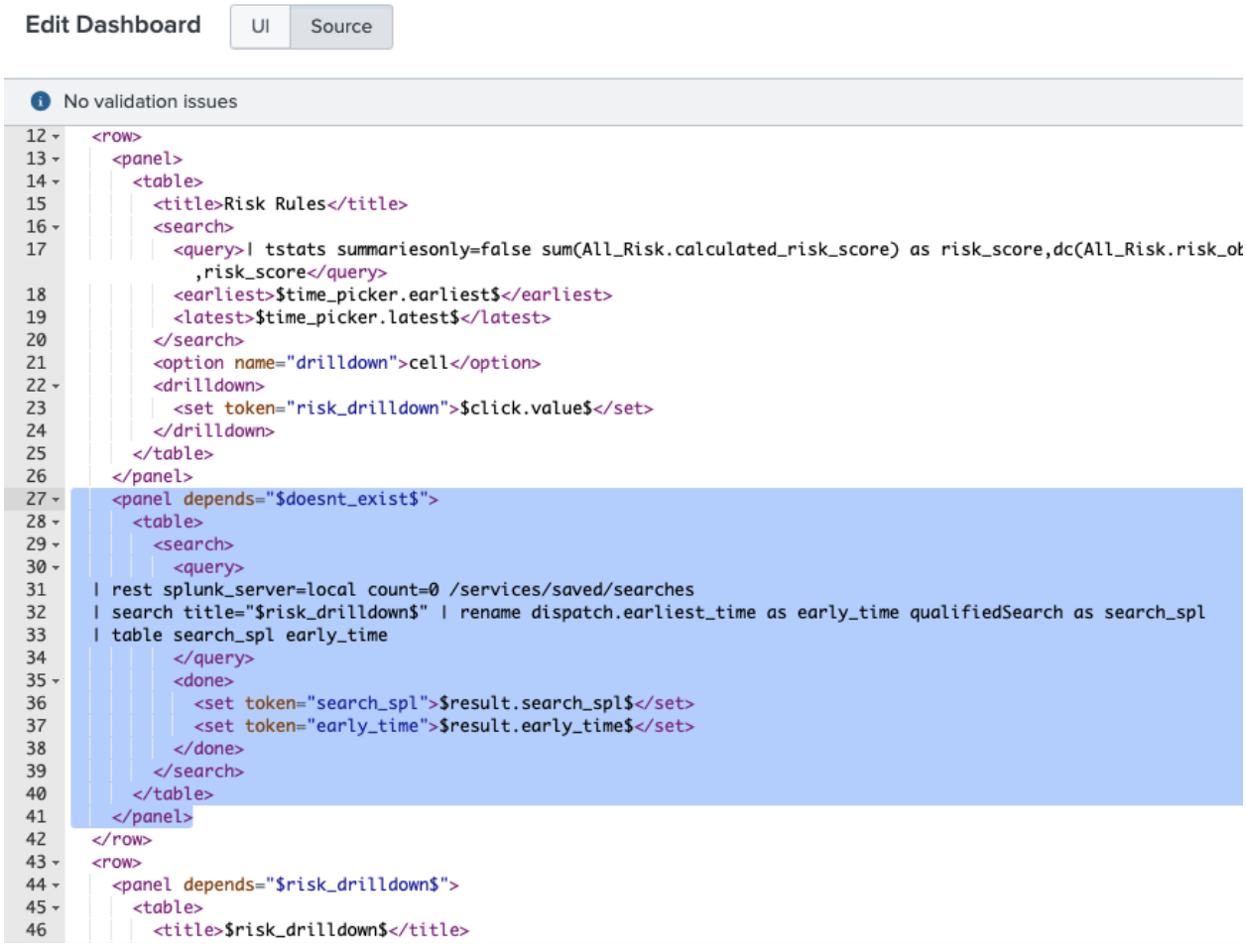
Next, open a drilldown view in a new tab with threat object column clicks.	`<drilldown>`
	`<link target="_blank">`
	`$click.name2$`

Preserve the ability to dismiss the panel.	`<unset>`

Be able to click a threat object to open a new tab with correlation search results or anywhere else to dismiss the panel.	`$threat_object$`
	`<drilldown>`
	`<condition>`

Let's start with adding a search which runs in the background to pull out the correlation search SPL. I'm going to keep it in the same `<row>` as the initial search in a new `<panel>` that looks for non-existent token `$doesnt_exist$` so it can run without being seen:

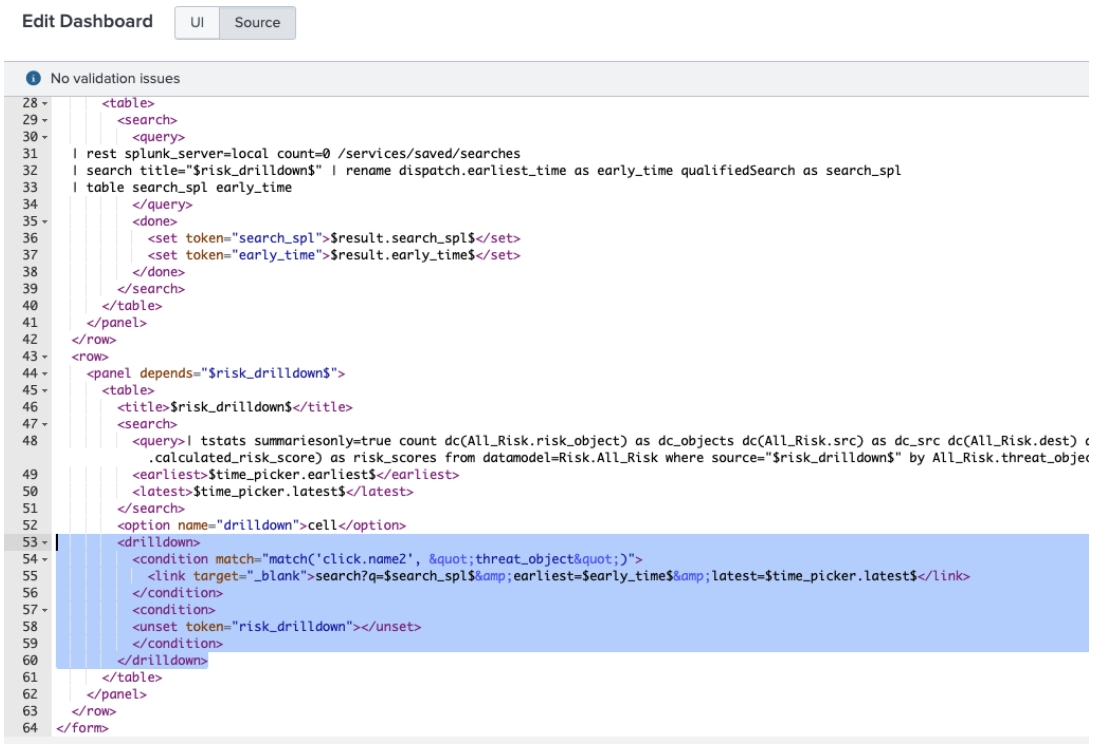
```
    <panel depends="$doesnt_exist$">
      <table>
        <search>
          <query>
| rest splunk_server=local count=0 /services/saved/searches
| search title="$risk_drilldown$"
| rename dispatch.earliest_time as early_time qualifiedSearch as search_spl
| table search_spl early_time
          </query>
          <done>
            <set token="search_spl">$result.search_spl$</set>
            <set token="early_time">$result.early_time$</set>
          </done>
        </search>
      </table>
    </panel>
```



This example uses the REST command to look at the actual correlation searches and pull out the SPL and its earliest running time, but you could bring along all sorts of nifty information with that base search. Then you can use `<done>` so when this search finishes running, you can grab what's in the `$result.fields$` to populate the tokens `$search_spl$` and `$early_time$` to use in the new drilldown.

Now, add some `<condition>` logic so you can differentiate between a click on the `threat_object` field for digging into events or dismissing the panel. Try adding this into the `<drilldown>`:

```
<drilldown>
  <condition match="match('click.name2', &amp;quot;threat_object&amp;quot;)">
  <link
target="_blank">search?q=$search_spl$&amp;amp;earliest=$early_time$&amp;amp;latest=$time_picker.l
atest$</link>
  </condition>
  <condition>
    <unset token="risk_drilldown"></unset>
  </condition>
</drilldown>
```



What we're doing is utilizing `<condition>` as an if / else statement, but keep in mind you can have endless amounts of conditions. You could have different drilldowns based on each column they clicked, or even set a token based on a specific click within a multi-value field!

The match field within `<condition>` is a bit particular, which is why I enclose the field in single quotes and need to use percent-encoding for the match logic. Keep in mind you can use any regex you like in between the percent-encoded quotes, but you'll have to convert to percent-encoding to use it in Simple XML.

Go ahead and try it out!

I hope you're starting to see how building a logical stream of drilldowns in Splunk can make anyone's job so much easier. I also hope you start using this one to track down noise in your Risk Index! If you really want to dig into some wildness, check out Gabriel Vasseur's ES-Choreographer for monitoring and maintaining correlation searches, GV-Utils to add even more incredible dashboard functionality, and especially this blog post for some insightful best practices and tips and tricks with dashboards.

Appendix D: RBA Talks

Here is my full list of essential talks around RBA (as of September 2022):

Intrusion Detection along the Kill Chain: Why Your Detection System Sucks and What To Do About It

I love the title of this talk from John Flynn at Black Hat in 2012. Yes, TWENTY-TWELVE, folks! People have been solving this problem in their own silos and trying to talk about it, but there hasn't been any shared, concrete language around the methodology. I love that SPL makes this all so much easier, because so many tools just don't allow that much control over the data they present.

Tracking Noisy Behavior and Risk-Based Alerting with ATT&CK

Haylee Mills (that's me!) discusses the fundamental building blocks of RBA in any context. After talk Q&A with Cat Self.

When Insiders ATT&CK!

Matt Snyder from VMware digs into the details of building out Insider Risk based detections and alerts with RBA and MITRE ATT&CK. After-talk Q&A with Cat Self.

SEC1144C - Curating your Risk Ecology: Making RBA Magick

An essential talk by Haylee Mills (!) for learning how to tune your Risk Rules to produce high-fidelity events.

SEC1479 - Say Goodbye to Your Big Alert Pipeline, and Say Hello to Your New Risk-Based Approach

The original .conf talk by Jim Apger and Stuart McIntosh explains the approach and benefits.

SEC1803 - Modernize and Mature Your SOC with Risk-Based Alerting

Jim Apger reviews RBA structure and benefits, then Jimi Mills offers a detailed timeline of Texas Instruments' RBA evolution.

SEC1113A - Streamlining Analysis of Security Stories with Risk-Based Alerting

Haylee Mills (<3) explains how to design intuitive dashboards, and shows off what she built while at Charles Schwab.

SEC1908 - Tales From a Threat Team: Lessons & Strategies for Succeeding with a Risk-Based Approach

Stuart McIntosh of Outpost Security delivers handy lessons learned, metrics, and approaches from running RBA in production for over a year.

SEC1538 - Getting Started with Risk-Based Alerting and MITRE

Bryan Turner reviews RBA structure and benefits, then guides building detections and aligning to ATT&CK.

SEC1163A - Proactive Risk Based Alerting for Insider Threats

Incredible before and after metrics from Matt Snyder at VMware about how they revolutionized their Insider Threat program with RBA.

SEC1590C - Augmented Case Management With Risk Based Analytics and Splunk SOAR

Phil Royer and Kelby Shelton explain the Risk Notable SOAR content pack and how to deliver enrichment to Risk Notables.

SEC1708A - AbnorML Detections - Using Math To Stop Bad Guys in ES

Tyler Williams from SIAC discusses using ML models to create risk and find more needles in noise.

SEC1162A - Supercharge Your Risk Based Alerting (RBA) Implementation

Teresa Chila from Chevron goes over risk adjustment and tuning as well as how to design playbooks in SOAR with Risk Notables.

SEC1249A - Accenture's Journey to Risk Based Alerting with Splunk Enterprise Security and Beyond

Chip Stearns from Splunk and Marcus Boyd from Accenture discuss how RBA transformed their SOC, and the things to keep in mind while building it.

SEC1428B - Detection Technique Deep Dive

Doug Brown recaps and reframes some awesome statistical anomaly magick from his .conf17 talk. These are incredible events to have in RBA, and work well to enable UBA-esque functionality.

SEC1556 - Building Behavioral Detections: Cross-Correlating Suspicious Activity with the MITRE ATT&CK™ Framework

Haylee Mills (it me!) discusses the technical underpinnings of the original SA-RBA app's SPL, before RBA was part of ES 6.4.

SEC2186 - Pull up your SOCs 2.0

Dimitri McKay talks about the maturity of your SOC and what to really focus on. Love this breakdown and how RBA can enable so much of this process.