Introducing the Funnel of Fidelity

posts.specterops.io/introducing-the-funnel-of-fidelity-b1bb59b04036

December 3, 2019

At SpecterOps we work with companies to build robust detection and response programs. One of the stark realities is that human capital is limited, which is why we work with customers to find efficient ways to leverage their scarce resources. For instance, it is not feasible for an organization to review every single security event that is collected to determine if it is malicious. Our industry has many established processes to deal with this problem such as the tiered layout of a security operations center (SOC), however, we notice a fundamental disconnect between desired state and reality in organizations that we work with. As a result, I created a model to describe the conceptual process that organizations follow to quantify the high level roles and responsibilities of a detection and response program. As events pass through the model the depth of event analysis and fidelity is increased. For this reason I call the model the Funnel of Fidelity (following the naming convention of David Bianco's Pyramid of Pain).

Utilizing Limited Resources

In detection and response our goal is to intelligently identify where to apply our limited resources most effectively. This prioritization results in the creation of a funnel that smartly filters out noise and hopefully results in spending more time on the activity that is most likely to be malicious. The funnel consists of different stages that must be completed to successfully remediate an attack. Those stages, which will be described in detail below, are Collection, Detection, Triage, Investigation, and Remediation. Each stage is a critical component of a detection and response program. However, we often find that organizations emphasize certain stages differently and this leads to issues with the flow of the funnel.

Clogging the Funnel

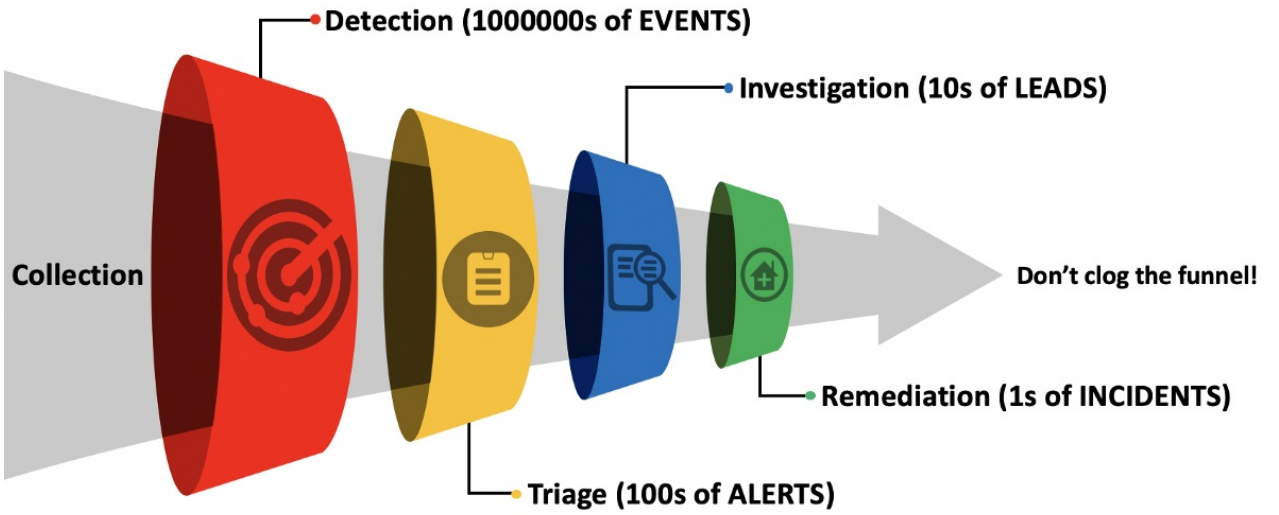
The idea of the funnel is that each stage (Collection, Detection, etc.) exists to filter out noise in a calculated way, but also affects the ability of a future step to be successful. I refer to this phenomena as a "clogged funnel". Lets explore some high level examples of clogging below:

- An organization does not have a SIEM with centralized telemetry collection. As a result, there aren't any events to build detections around which results in a clogged funnel.

- An organization has a functional level of centralized telemetry collection, but has not created any detections. As a result, there are no alerts being produced which limits the triage and investigation process resulting in a clogged funnel.
- An organization has great centralized telemetry collection, a robust detection engineering process, and really strong triage procedures; but they don't have an incident response plan. As a result, they can detect an attack, but cannot remediate it in a reliable way resulting in a clogged funnel.

Visualizing the Funnel

Below is an image of the Funnel of Fidelity. Imagine the size of the arrow as the quantity of inputs that must be addressed at each stage. Filtering occurs at each of the colored rings to reduce the magnitude of generic events that are passed to the next step of the process.



Stages

The Funnel of Fidelity depicts the process of applying different analytical procedures to manage millions of contextual events and apply limited investigative resources to the events or situations that are most likely to be malicious. The funnel consists of 5 stages: collection, detection, triage, investigation, and remediation. Each stage takes an input that was generated in the previous stage, performs some sort of filtering or noise reduction, and produces an output for the following stage. Ideally, each stage allows for deeper or more manual analysis to be applied to the event in question because non-relevant events have been filtered out.

The remainder of this post describes the stages in depth to give an idea of what is involved with each stage and how they interact with each other. For the sake of clarity, the input and output of each stage is named specifically (e.g., events, alerts, leads, and incidents). Additionally, each stage description includes the stage's responsible role, input, and output.

The roles specified below are representative of conceptual responsibilities within an organization. Organizations will staff these roles in different ways depending on manpower and organizational constraints. For instance, there may not be an individual who is explicitly responsible for each of the described roles. We commonly see individuals who wear numerous hats as part of their responsibilities (e.g. Tier 3 SOC Analyst who is responsible for Detection Engineering, Alert Triage, and Investigation).

Collection

- Role: Data Engineer
- Input: Data Sensors
- Output: Events

Telemetry is the building block of security monitoring as it provides context about activity occurring throughout the environment. Without telemetry it is nearly impossible to detect malicious activity. Mature organizations should strive to centralize as much telemetry as possible to enable enterprise detection activities. For example, Windows Event Logs are generated and stored locally, but must be centralized to support enterprise detection efforts. The collection phase gathers events from data sensors (Windows event logs, commercial EDR solution, proxy logs, netflow, etc.) and makes them available for the detection stage.

Commonly, detection and response teams look to the collection phase as the problem point because they assume that their problems arise from a lack of robust telemetry collection which clogs the funnel early on. The reality is that most organizations have a decent base level of collection that can be improved, but also provides enough telemetry to get started. It is uncommon for us to interact with a client that is leveraging collected telemetry to its fullest extent.

Detection

- Role: Detection Engineer
- Input: Events
- Output: Alerts

With events being collected in a centralized fashion, detection engineers define detection logic to identify events that are security relevant. The goal of the detection stage is to reduce the millions or billions of events from the collection stage to hundreds or even thousands of alerts which will be analyzed during the triage stage.

These detections are created in an interactive process often referred to as threat hunting or detection engineering, but should be implemented in production through an automated process where detection logic is applied to events to generate alerts. Generally we see that the detection stage converts millions of events generated through collection into hundreds of alerts which are passed on to the triage phase.

Triage

- Role: SOC Analyst (Tier 1 or 2)
- Input: Alerts
- Output: Leads

Alerts are the result of detection logic, but it is reasonable to expect some amount of false positives. The triage stage is where SOC analysts work to categorize alerts as known bad (malicious), known good (benign), and unknown activity. Malicious activity is immediately identified as an incident and moved to the remediation stage, while unknown activity is identified as a lead and sent to the investigation stage as it requires additional scrutiny.

The triage stage is where we see many organizations struggle. This typically manifests itself in malicious activity being marked as a false positive through alert fatigue. It is very common for organizations to delegate triage responsibilities to tier 1 SOC analysts, without checks and balances to ensure success. A large contributing factor to alert fatigue is the often unclear nature of alerts in general. SOC analysts have a hard time understanding the goal of the detection logic that produced an alert, the context of the attack that is being detected, and the steps they should take to properly triage the alert. These issues can be mitigated by using a detection documentation standard like Palantir's Alerting and Detection Strategy Framework.

Investigation

- Role: SOC Analyst (Tier 2 or 3) or Forensic Analyst
- Input: Leads
- Output: Incidents

The triage stage works to remove false positives from the pipeline and results in a manageable number of leads (likely in the single or double digits). A lead is an activity that cannot be identified as malicious or benign and thus requires additional investigation. The investigation stage is used to collect additional context that may not be available during the detection or triage phases. This may involve more manual / less scalable analysis such as file system analysis, memory forensics, binary analysis, etc. to help identify the true source of the activity. This additional scrutiny is possible because of the reduction in noise that occurred during the previous stages.

Remediation

- Role: Incident Responder
- Input: Incident
- Output: N/A

Once an incident is declared, remediation activities must occur. This is the phase where incident responders work to identify the scope of the incident and remove the infection

from the network. Many organizations work with third parties to accomplish remediation activities and ensure that they are completed in a timely manner. It is important to practice remediation in non-emergency situations to ensure the plan is sufficient and any issues are worked out.

What is Detection?

The concept of detection tends to be very nuanced in many organizations. For this reason we must distinguish between micro detection (the process of writing logic to alert on a potentially malicious event) and macro detection (the process of taking a true positive event from alert all the way to remediation). To truly consider an attack as detected, in the macro sense, the attack must result in remediation activity of some kind. Anything less is considered passive detection, which in the grand scheme of detection and response doesn't matter. Below, I will explore two example cases where I've seen confusion regarding the concept of detection.

Example 1

Red team assessments often include a debrief of the attack path for the defenders. Commonly, during the debrief, someone on the detection and response team will learn about an attack that was carried out by the red team and will begin reviewing events in their SIEM. This exercise frequently concludes with the defender saying something along the lines of , "yea we saw that... here is the event". This exercise is valuable, but what the defender is actually saying is "yes we collected relevant information to that attack, but we have not yet created the detection logic to detect that activity".

Example 2

We've seen numerous organizations that have detection logic built for a specific technique that we used during a red team exercise, for example Kerberoasting, but for some reason the SOC never detected the activity. We eventually find that an alert fired, but it was marked as a false positive. Unfortunately the analyst responsible for this ticket didn't know enough about Kerberoasting to differentiate between benign and malicious service ticket requests. The activity may have alerted, but in reality there was no detection that occurred in the macro sense.

How can I use the Funnel?

A huge benefit of the Funnel of Fidelity concept is that we can diagnose at which stage the breakdown is occurring. In example 1, it appears that there is sufficient collection, but a robust detection is missing. We could work with this customer to identify strategies to engineer robust detections using the data they are currently collecting in their environment. In example 2, we see that an alert is produced, but the triage process is failing. To address this we could focus on building alert documentation to grow

organizational knowledge and remove guess work from the alert triage process. Both of these examples should not be seen as a failure of the detection and response program. Instead they should be viewed as an opportunity for process improvement (which is a great topic for future posts).