Priam Cyber AI

The root of all evil are alerts...

Introduction

This is part I of a series of technical papers discussing the challenges of alert management in security operations.

Problem

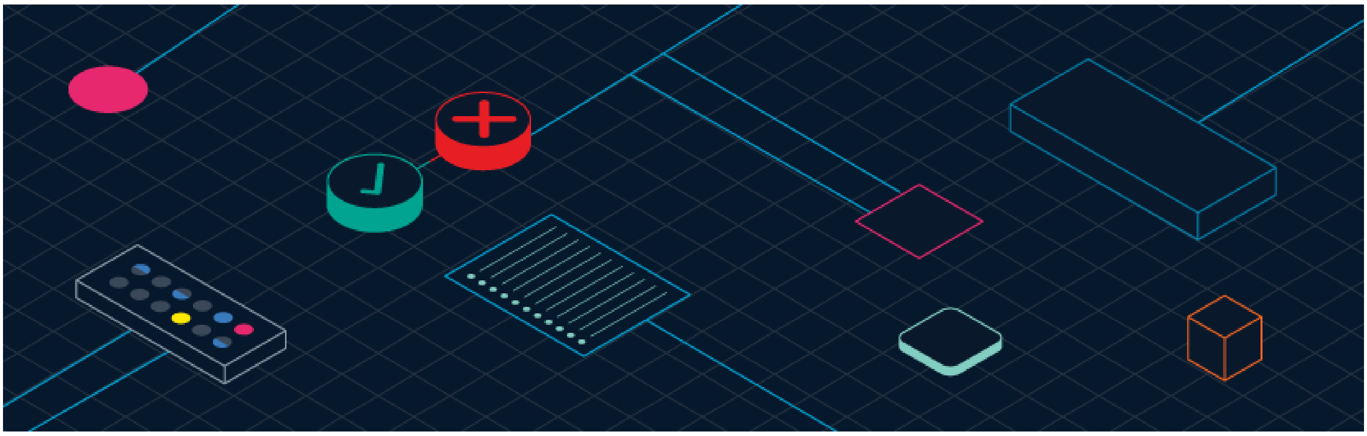
False positives - alerts that incorrectly indicate a security threat- are a major problem for security operations centers (SOCs). Numerous studies have shown that SOC analysts spend a disproportionate amount of time and effort triaging alerts that appear malicious but turn out to be benign in the end.
Each study is different because it targets a different sample of the population but it is worth mentioning some key stats from the most popular reports below:

Research that Invict ("Security teams suffer from alert overload") conducted recently found that SOCs waste an average of 10,000 hours and some $500,000 annually on validating unreliable and incorrect vulnerability alerts.

Another survey ("Reaching the Tipping Point of Web Application and API Security") that Enterprise Strategy Group (ESG) conducted for Fastly found organizations reporting an average of 53 alerts a day from their web applications and API security tools. Nearly half (45%) are false positives. Nine in ten of the respondents in the survey described false positives as having a negative impact on the security team. Furthermore 75% of the surveyed businesses spent as much, or more, time chasing false positives than they did dealing with actual security incidents.

In another report ("70% Of SOC Teams Emotionally Overwhelmed By Security Alert Volume") from Trend Micro which polled 2,303 IT security and SOC decision makers across companies of all sizes and verticals, 70% of respondents say they are stressed by their work managing IT threat alerts. This comes as the majority (51%) feel their team is being overwhelmed by the volume of alerts and 55% admit that they aren't entirely confident in their ability to prioritize and respond to them. It's no wonder therefore that teams are spending as much as 27% of their time dealing with false positives.

Ponemon Institute did a report with Exabeam ("The Exabeam 2019 State of the SOC Report"), revealing that on average, security personnel in U.S. enterprises waste approximately 25 percent of their time chasing false positives because security alerts or indicators of compromise (IOCs) are erroneous. The report also highlighted the need for security operations center (SOC) productivity improvements, citing that security teams must evaluate and respond to nearly 4,000 security alerts per week.

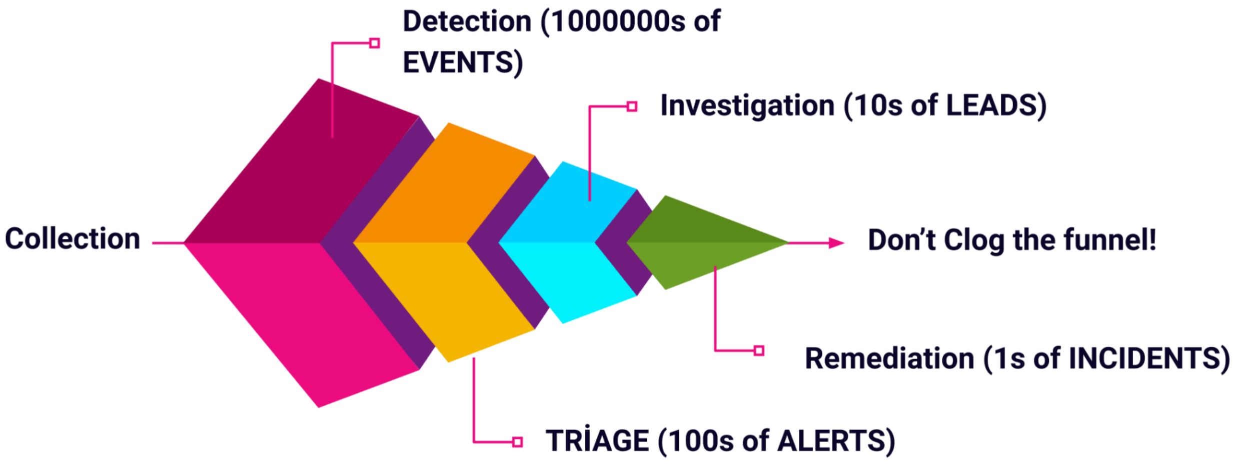


The one million dollar question then becomes: how do we reduce false positives?
First of all let’s take a step back and define some main concepts borrowed by the field of statistics and machine learning. Second, we can define guiding principles and a framework that we can adopt to improve this problem. Third, we introduce a framework called SigmaTau -an active project from Priam- that will benefit the wider community.

Operational context

The funnel of fidelity was a concept introduced by SpectreOps in this article.

The figure below shows the Funnel of Fidelity: the size of the arrow shows 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.


Detection (1000000s of EVENTS)
Investigation (10s of LEADS)
Collection
Don’t Clog the funnel!
Remediation (1s of INCIDENTS)
TRİAGE (100s of ALERTS)

“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.”

The 5 Whys

We didn’t invent this methodology, Five whys (or 5 whys) is an iterative interrogative technique used to explore the cause-and-effect relationships underlying a particular problem. The primary goal of the technique is to determine the root cause of a defect or problem by repeating the question "Why?" five times.

The number ‘5’ here comes from the anecdotal observation that five iterations of asking why is usually sufficient enough to reveal the root cause.

In some cases, it may take more or fewer whys, depending on the depth of the root cause. If you have children you know how frustrating it can be when challenged about your assumptions and at some point you have to stop when you either stop at a dogmatic or axiomatic fact.

The Five Whys method was originally developed by Sakichi Toyoda, the founder of Toyota Industries. This method became widely used in Toyota Motor Corporation and is still used frequently to this day. Taiichi Ohno, the architect of the Toyota Production System, describes the five whys as “the basis of Toyota’s scientific approach.”

Along with their other “go and see” philosophies, the five whys method is used along with other famous concepts such as kaizen, poka-yoke, and jidoka which we will introduce in a future article.

We found that this method -which has been also used in other fields of computer science and business- is particularly useful for security operations when diagnosing the quality of the detection events coming from various products.

Since we have been using this technique to our benefits we would like to share our findings when triaging alerts in real operations:

1. Metrics

The rule doesn't provide sufficient or statistical metrics such as true positive rate, false positive rate or "confidence".

2. Categorization

What part of the attack chain the rule is supposed to detect? What framework is following for example the diamond model or the ATT&CK chain.

3. Parametrization

What are the parameters of the rule that should be changed to fit your environment?

4. Modularity

Could we combine a rule with others and how? How are those metrics affected?

5. Accountability

Who wrote the detection, when and why? What was the context of the incident?

6. .Binary vs Fuzzy

The nature of a detection whether is binary (with a parameter threshold or without a parameter) or fuzzy in nature with a categorical output (classification) or severity (likelihood) score .

7. Forensic

What additional forensics evidence should be gathered during triage?

The best use of this list is a checklist: if the cause is in that list you should go and fix the problem otherwise you should apply the 5 whys to determine the root cause analysis.

In this article we will focus on the first cause related to metrics definition.

ML concepts

First we will revisit the concepts of precision, recall, true positives, false positives, false negatives, true negatives.

To illustrate all those concepts we will assume a fictitious SOC team that receives alerts from a variety of cyber security products. The type of product does not matter at all but it is defined by certain properties which are usually probabilistic in nature. We assume that the cyber product is producing a binary decision of malicious vs benign outcome from some event logs. This is of course a simplification but most products nowadays offer a binary decision mode but also a more fuzzy approach if enabled.

Precision vs Recall:

Precision and recall are the measures used in the information retrieval domain to measure how well an information retrieval system retrieves the relevant documents requested by a user.

The measures are defined as follows:

$$\text{Precision} = \frac{\textit{Total number of documents retrieved that are relevant}}{\textit{Total number of documents that are retrieved}}$$

$$\text{Recall} = \frac{\textit{Total number of documents retrieved that are relevant}}{\textit{Total number of relevant documents in the database}}$$

Example: a detection engine has a precision of 90% and a recall of 10%, therefore it is expected that on 100 returned matches, 90 of them will be valid malicious. However assuming that there are 10 intrusions (unknown to the SOC team of course!), only 10% which is 1 in this case will be identified leaving 9 unreported intrusions.

Precision and recall are not particularly useful metrics when used in isolation. For instance, it is possible to have perfect recall by simply tagging every event as malicious. Likewise, it is possible to have near-perfect precision by selecting only a very small number of malicious events. This is why you could calculate a derived metric called the F-measure which is simply:

$$F = 2 * \frac{precision * recall}{precision + recall}$$

This suggests a few rule of thumbs when selecting a product:

- ➔ If a vendor only reports either precision or recall then is probably trying to hide the true performance
- ➔ If a vendor reports a high precision and high recall then validate how realistic was their test set like a public test set vs a private test set
- ➔ If a vendor reports a high precision and low recall, or high recall and low precision than seems more like a legitimate claim

Precision and recall can be calculated with the following formulas:

- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)

Where TP = true positive, FP = false positive, FN = false negative which will be described in the confusion matrix later on.

This concept of precision vs recall tradeoff was also explored by SpectreOps but with a different terminology in relation with the figure below.

Precise

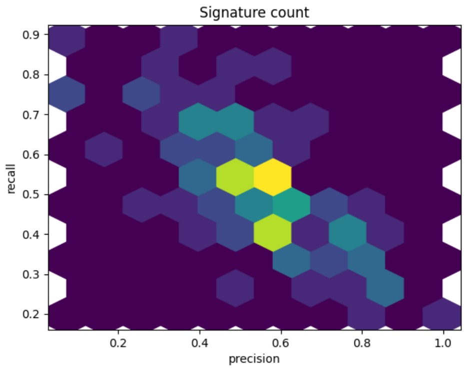
Low FP
High FN

Broad

High FP
Low FN

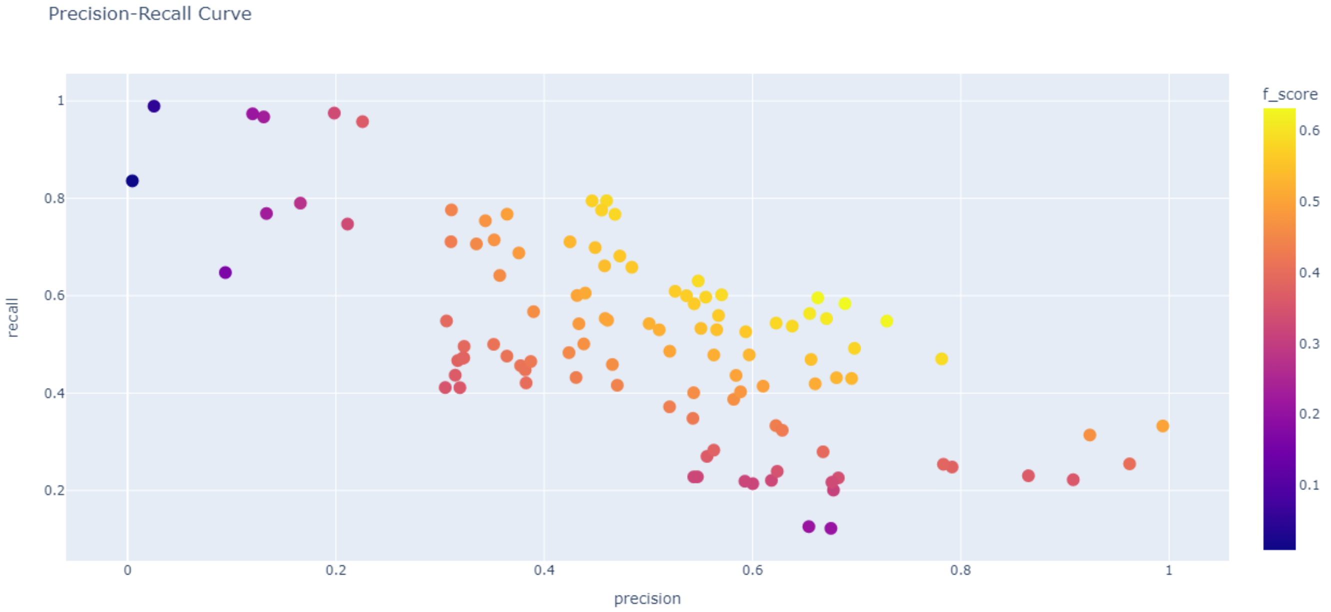
On the left side you have a detection that has high precision and low recall, on the right side you have a detection that has a high recall with a lower precision presumably based on the fact they have identical true positives.

It is a good intuitive explanation of the tradeoff but we prefer to adopt the precision, recall and F measure formulas because they have the advantage of being normalized between 0 and 1 as rational numbers.

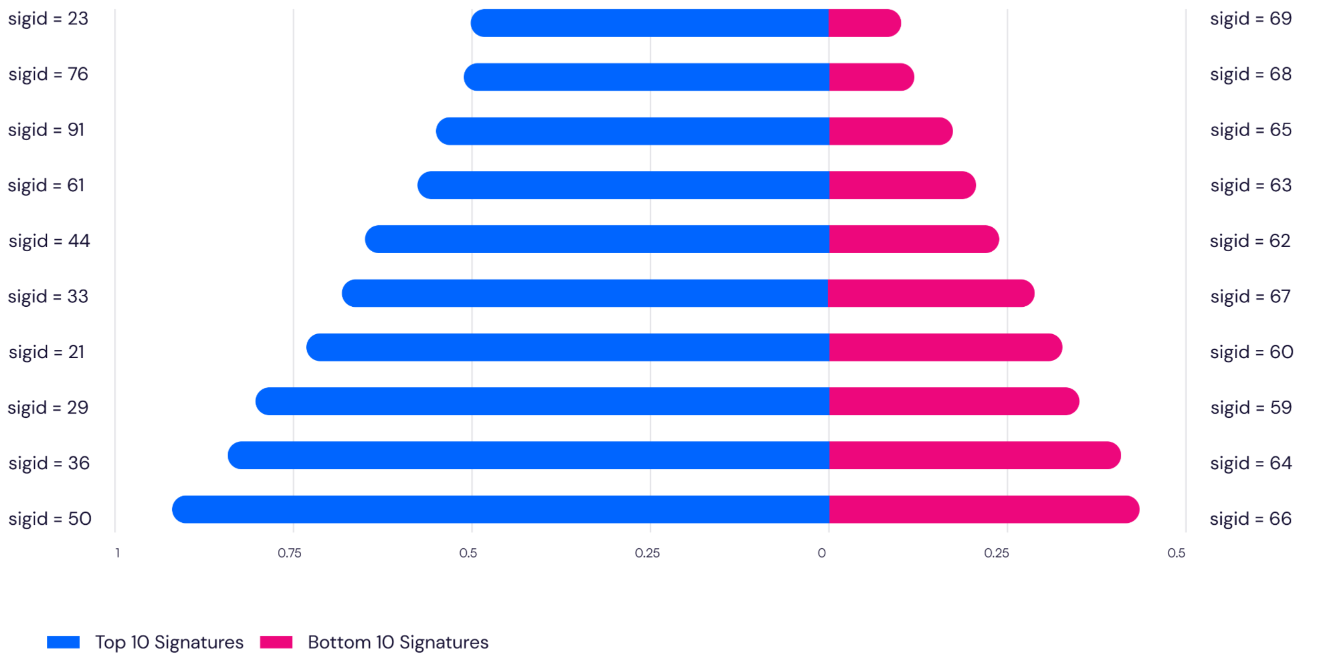


We found it most effective to use scatter plots and hexaplots to plot the distribution of scores in our SOC platform like in the following pictures on the left.

The challenge is then how do we measure those quantities from a subjective perceived scale to an objective metric.



Top and Bottom 10 Signatures



To do so let's expand more on the meaning of those quantities with a concept from machine learning called the confusion matrix which is suitable for binary detections. Please note that for ternary or multi class detections this matrix needs to be expanded and we will describe that in future articles.

A confusion matrix for binary detections looks like this:

Total Population	An intrusion	No intrusion
An Alert Is Generated	True Positive (TP)	False Positive (FP) or Type I error
No Alert Is Generated	False Negative (FN) or Type II error	True Negative (TN)

Looking back on the precision and recall formulas, we can see that precision relates to the Type I error and the recall relates to the Type II error.

Most vendors usually reports only the accuracy metric:

$$\frac{TP+TN}{TP + TN + FP + FN}$$

which includes both the Type I and Type II errors, **this metric can be deceiving in most cyber domains where the distribution of malicious vs bening activity is very skewed.** Consider a sample with 95 benign samples and 5 malicious samples. Classifying all values as positives in this case gives a 95% accuracy score.

How do we calculate all the 4 introduced variables from precision and recall? If you remember your algebra, unfortunately we cannot because we have only 2 equations in 3 variables so the system is underspecified.

Cost of calculating metrics

The TP and FP numbers can easily be calculated during your security operations, this is true even when your soc team is using third party signatures. Every time you do a triage session you will typically tag an alert as a true positive/false positive or undetermined (this will introduce uncertainty in the matrix which we will discuss later). You can then calculate the ratio based on the total amount of alerts you triaged during that session.

For example let's assume your team reviewed 100 alerts in a day coming from a specific playbook, TP = 10 of them were true positives and FP = 90 were false positives, then the precision will simply be: (10)/(10+90) = 0.1 that is 10%.

Now it is important to note that this precision will be a function of the time: precision(t) because each time it is calculated we are looking only at a sample of the entire population. There is a very desirable property of sampling called i.i.d. (which we will discuss in a future article) that will allow us to estimate the precision of the population over time by introducing a confidence interval to the measure of precision and recall. The confidence interval here is a statistical concept and is not the usual interpretation of the confidence included in signature descriptions.

The second measure to estimate is now recall, but this introduces a problem: how do we estimate false negatives? We need to have prior knowledge of an intrusion which is the entire point of having an incident response team. Where do you find an oracle then?

There are 2 possible solutions to this:

- ➔ Historical analysis: you can retrospectively review your past incidents in terms of forensics artifacts and identify which signatures/detections didn't trigger an alert
- ➔ Simulation analysis: you can run a breach attack simulator on your infrastructure or in a cyber range, then measure both the precision and recall of the entire signatures.

The historical analysis approach is more expensive in terms of implementation but has the advantage of being more high fidelity, the simulation analysis is less expensive in terms of implementation but has the disadvantage of being lower fidelity.

The two approaches can be mixed -however- attention should be paid when correlations are present requiring the estimation of a covariance matrix for the confidence estimates.

Let's assume for simplicity that you then conduct a breach attack simulation with a free tool such as Caldera or InfectionMonkey and you receive 100 alerts from your playbook, this time you know that there are 20 events in total that should be detected as malicious.

You identify 10 -as before- as true positive but there are 10 missing which should be there from your ground truth.

Then your metrics become:

- precision = (10)/(10+90) = 10 %
- recall = (10)/(10+10) = 50 %
- F-measure = 2 * (0.1* 0.5) / (0.1 + 0.5) = 0.16 = 16 %

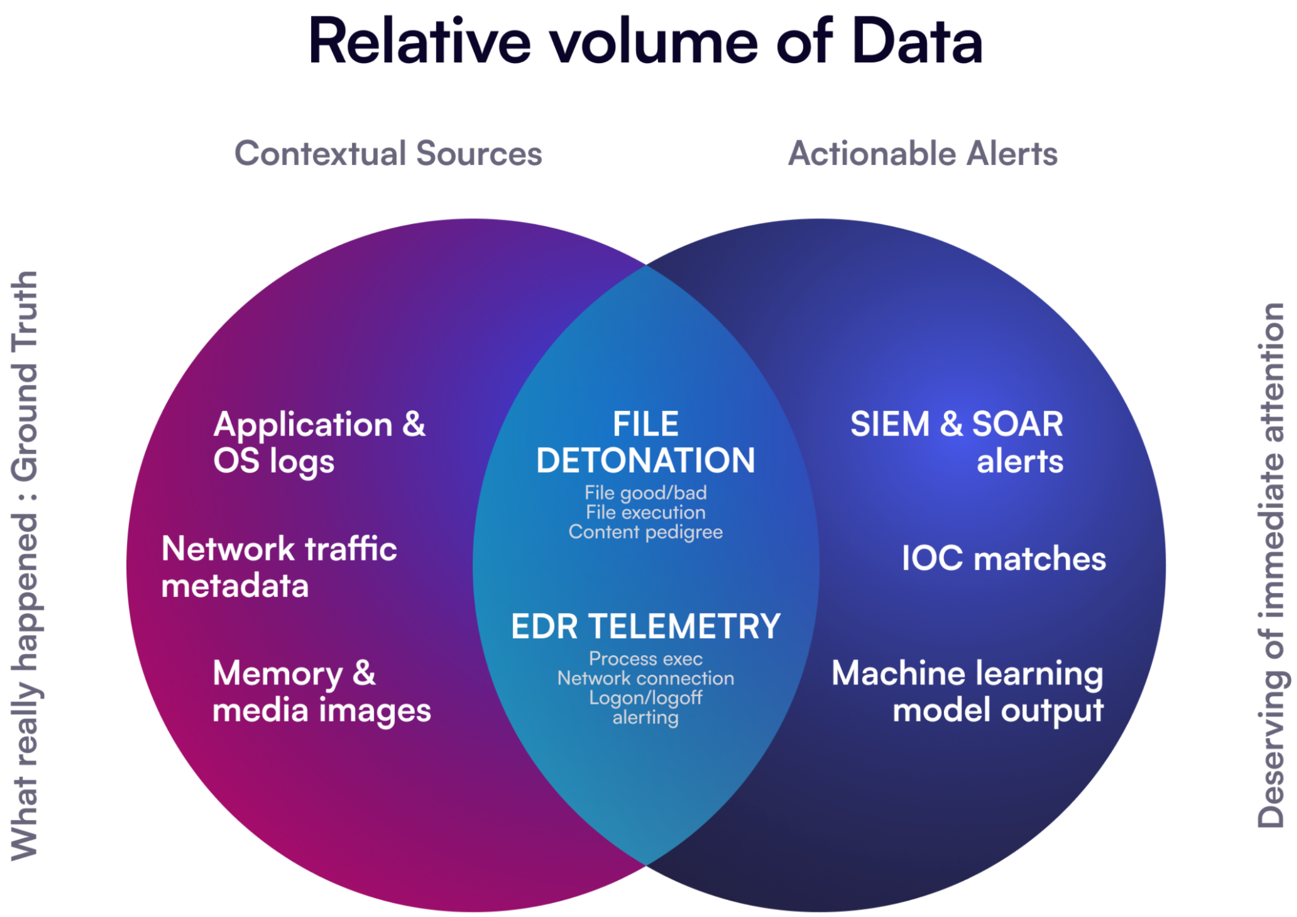
You can then repeat the same process for all your playbooks or detection products and plot the distribution for precision, recall and F-measure of your entire dataset. You should also measure the time (cost) it takes to investigate each alert category vs the total volume of alerts.

The resulting table can then be used for the following downstream tasks:

- ➔ Disable low performing detections: when a detection lack tuning parameter and it has a high cost of triage then it could be disabled
- ➔ Improve detections: when a detection includes a tuning parameter, you can optimize it to achieve a desired precision, recall or f-measure target

- ➔ Avoid human bias: displaying those metrics for each alert will avoid biasing your security team based on their subconscious estimation of FP rates.
- ➔ Improve prioritization: alerts can be sorted in descending order with the F-measure because those with a high value are very likely to indicate a real attack. Furthermore the cost of investigation will usually correlate with the metric: higher cost with high recall values vs lower cost for high precision. A regression analysis can possibly predict the triage cost of other similar detections.

The cost of triaging an alert is usually directly proportional to the amount of contextual data required as described in the book (Parker et al. 10) where the relative amount of data increases when investigating the contextual source of an alert.



Ranking is a complex topic because it needs to take into account also the asset criticality, the cost of triage, the potential impact and thus will be discussed in the future sessions.

The human bias is one of the most difficult to grasp because even most senior security operators are unaware for example of the base rate fallacy which we want to describe in the next section since it is one of the most important ones.

There are also other cognitive fallacies which we will describe in future papers such as the availability bias.

The Base Rate Fallacy

The base-rate fallacy is a cognitive bias that leads people to make inconsistent and illogical decisions.

It occurs when individuals overweight or ignore information about the probability of an event occurring, in favor of information that is irrelevant to the outcome.

This cognitive bias can lead to irrational decisions and behavior. For example, if someone were told that one person among a group of 100 had contracted a fatal disease, they may be more likely to go see their doctor for routine checkups.

This, although not seeming like an outwardly harmful action, could lead to a cumulative overburdening of the healthcare system, and thus various negative effects.

The same principle applies to security operations, let's make a fictitious example where we have regular activity (R events) and intrusion activity (I events).

An EDR event will ring an alarm for 99 out of 100 **I events** (precision = 99/100) but for only 1 out of 100 **R events** will ring a false alarm.

The matrix looks like this now.

Total Population	An Intrusion	No Intrusion
An Alert Is Generated	99	1
No Alert Is Generated	?	?

Suppose further that we have a computer system on which 1,000,000 *R events* take place in an hour plus 100 *I events*. This means that the benign activity is 10000x the malicious activity which is a fairly reasonable assumption.

We observe an alarm from the EDR: what is the chance that this event is malicious?

Answer: The system will ring an alarm for 99 of the *I events* and for 10,000 of the *R events*. Given an alarm, the chance that this is an intruder event is 99/(10,000+99) which is roughly 1 percent.

So the chance of a false alarm is 99% which is surprisingly high compared to 99% of precision.

The table below shows how the false alarm rate changes with the percentage of benign activity.

Precision	I/R Rate	True Alarm Rate	False Alarm Rate
99%	10000x	0.0098	0.9902
99%	1000x	0.0901	0.91
99%	100x	0.4975	0.5
99%	10x	0.9083	0.09
99%	1x	0.9900	0.01

The table shows an important effect when the intrusion rate is high which is particularly the case for products such as IPS/IDS/EDR where the volume of benign events is very high compared to malicious events, the true false alarm rate will kill your SOC performance because your team will start to ignore those alerts.

Let's make one more example with a spam filter product with a declared accuracy of 99%: 99% of all spam messages are classed as spam, while 99% of not-spam is not tagged.

On an average day your mail server receives 1,000,000 messages, of which 5% is actually spam and the rest isn't.

With this information we can easily build a full confusion matrix below:

Alerts	Spam (I)	Not Spam (R)	Total Samples
Spam Alert	49,500	9,500	59,000
No Alert	500	940,500	941000
Semi Totals	50,000	950,500	1,000,000

What is the likelihood that a mail will be really spam when there is an alert?

Looking at the first row the true positive rate is: 9,500/(59,000) = 0.16 which is around 16%.

Therefore to be resistant to the base rate, an ideal product or detection should have

- Precision = 1.0
- Recall = 1.0
- False Positive Rate = 0.0

This is impossible to achieve in practice for example even with a perfect precision and recall, if we want to achieve a true positive rate of more than 0.5 (50%), we would need to reduce the False Positive Rate under 0.00001 which is quite a challenge believe me!

Therefore we should keep in mind the following:

➔ Precision and Recall are as much important as the False Positive Rate

- ➔ The true false positive rate will depend on the state of your environment: when there is a lot of malicious activity it will decrease, when there is mostly benign activity it will decrease
- ➔ Alert fatigue is a direct effect of the base rate fallacy

Our approach to alert fatigue

At Priam Cyber AI we are tackling alert fatigue with the following strategies:

- ➔ Reducing the cost of triaging alerts via automated reinforcement learning bots
- ➔ Continuous estimation of precision/recall/f-measure in our platform
- ➔ Automatic optimization of playbooks and detections via genetic algorithms
- ➔ A framework called Sigma Tau to share detection metrics between users

As the cyber security issues and attacks are growing, security precautions are growing and so more solutions are being developed - in the macro categories of SIEM, SOAR, EDR, MDR, XDR etc solutions- in order to secure the environments and manage the security incidents. However the alert volume is also growing -some argue exponentially with complexity- and managing the incidents is becoming more problematic and time consuming. Therefore security teams are struggling to investigate increasing volume of complex alerts so they need to buy more time.

This has a direct financial impact on the cost of human labor. Our approach is not to remove the human from the equation but to simplify the information stream for operators to be most effective and spend time only on the most creative tasks such as threat hunting. This means operators are less stressed, making less mistakes and focusing on being more proactive instead of reactive. As a result they will have more time to spend on prevention whilst AVA will learn to automate boring tasks such as triaging and mitigation/remediations.

Finally our approach will improve their skills set and adaptively train the workforce against new emerging threats..

Sigma Tau

The Sigma Tau standard is a schema extension to the Sigma standard that allows publishers to include the metrics introduced in this paper.

Sigma is a generic and open signature format that allows you to describe relevant log events in a straightforward manner. The rule format is very flexible, easy to write and applicable to any type of log file. The main purpose of this project is to provide a structured form in which researchers or analysts can describe their once developed detection methods and make them shareable with others.

The project is hosted on this github repository and we welcome feedback and contributions from the community.

References

Arp, Daniel, and Erwin Quiring. "Dos and Don'ts of Machine Learning in Computer Security." 10 8 2022, https://dodo-mlsec.org/. *Accessed 16 1 2023.*

"The Exabeam 2019 State of the SOC Report." *Exabeam,* https://www.exabeam.com/library/2019-exabeam-state-of-the-soc-report/. *Accessed 16 January 2023.*

Parker, Ingrid, et al. *11 Strategies of a World-Class Cybersecurity Operations Center*. MITRE, 2022. *Accessed 16 January 2023.*

"Reaching the Tipping Point of Web Application and API Security." *Fastly,* https://www.fastly.com/web-application-and-api-security-tipping-point. *Accessed 16 January 2023.*

"Security teams suffer from alert overload." *Analysis of over 500K vulnerability reports over six years indicates enterprises can save 9,760 hours and $488,000 annually,* https://www.invicti.com/clp/proof-based-scanning-whitepaper/. *Accessed 16 January 2023.*

"70% Of SOC Teams Emotionally Overwhelmed By Security Alert Volume." *Trend Micro study reveals the human cost of underpowered Security Operations Centers*, 25 May 2021, https://newsroom.trendmicro.com/2021-05-25-70-Of-SOC-Teams-Emotionally-Overwhelmed-By-Security-Alert-Volume. *Accessed 16 January 2023.*