# Understanding signal-to-noise for vulnerability management success

BY ELEANOR BARLOW · NOVEMBER 25, 2025 · LAST UPDATED ON DECEMBER 24, 2025

## What you will learn

- **What signal-to-noise means in vulnerability programs:** Learn how to measure the value of vulnerability reports versus low-quality noise to focus on what truly matters.
- **How to improve your program's signal ratio:** Understand how scope, policy, rewards, and processes affect the quality of submissions you receive.
- **Practical steps to enhance outcomes:** See what changes help raise meaningful findings and reduce wasted effort in vulnerability management.

A common worry for IT and security teams is that, when operating an effective vulnerability management model, they will be flooded with potential vulnerability reports they likely don't have the capacity to work through.

But the real issue here is not volume; it's noise. Invalid or low-quality submissions can drain resources, cover up, or deprioritize critical signals that have real business impact, and demoralize the operational teams to the point of believing they won't find value in the volume of reports.

Trying to figure out what actions would raise the signal score, or revitalize a dying program, can be overwhelming. Intigriti's triage team filters out the gems within this noise, ensuring organizations categorize correctly, to see their signal-to-noise ratio regarding submission levels.

'Without a dedicated team focused on maintaining a program, the volume of work for handling reports can quickly become overwhelming. There are often a fair number of low-quality or false-positive reports that can take time to sift through, and in the absence of a triage team, these kinds of submissions cannot be differentiated from others, meaning every report is potentially taken as a serious threat until it is proven otherwise.' – Bug bounty DIY

**This blog looks at how to turn your signal-to-noise ratio into a key metric to support your cybersecurity journey. We explore what this ratio means, how to score it, and look at common challenges companies face regarding scope, policy, staff, rewards, researchers, and processes.**

## What does signal-to-noise mean?

To kickstart this discussion, **Chris Holt, Strategic Engagement and Community Architect** at Intigriti, provided his description of what signal-to-noise ratio means.

> **“The signal-to-noise ratio, from a program perspective, is a very simplistic way to measure program value. This is not to be confused with ROSI; it is not a metric to showcase Return on Security Investment, but rather a metric to show if the program is producing value to the company. A higher program signal score means, for any given report, a much greater likelihood that it has high value. Therefore, the resources taken to process that report are well spent because it 'moves the needle', so to speak, for the organization or product security. ”**
>
> **Chris Holt, Strategic Engagement and Community Architect, Intigriti**

# How to classify signal-to-noise to support impact?

Signal is the set of reports that provide value to the organization, and each program must define its own threshold value. In the simplest form signal can be defined as **reports that received a payment**, and would include submissions identified as valid, original (non-duplicate), and within program scope and quality requirements. These paid reports result in real value impact, reducing risks and/or enhancing security posture.

A more inclusive definition for signal can also include reports marked as an **“accepted risk”**, which are issues that are real, but the organization chooses not to fix, and **“rejected as a duplicate”** reports that describe a legitimate vulnerability already known to the team but was not the first to uncover the issue. While not rewarded, these still reflect the correct security-minded thinking and testing by security researchers. These two classifications typically do not receive a payment and thus would not appear in the most basic definition of signal.

**Noise, on the other hand,** refers to submissions that do not provide meaningful security value and do not lead to a bounty payout. These can include non-exploitable findings or low-quality reports with missing info/context/meaning with no actionable security issues.

Out of scope findings (n/a) that detail unrelated assets/program requirements are not considered for assessment or reward. Spam and AI-slop fall into the noise grouping quite easily. Brand new submissions are also placed in ‘Noise’ until triaged, as their value is unknown.

> **“Note that 'awaiting triage' and 'new' states will be classified as Noise in all cases, but they do not have a meaningful effect on the ratio, given how many reports sit in that pool at any given time.”**
>
> **Chris Holt, Strategic Engagement and Community Architect, Intigriti**

# How to score noise-to-signal. What are the indicators of low/high signal?

Think of your Signal and Noise as opposite values of a 100% sum. This means that if the Signal is 20%, the Noise must be 80%. First, count the number of valid reports and invalid reports, and then add them together for your total.

To work out the percentages, you would need to divide the number of valid reports by the total number of reports to get the Signal percentage. And then you would divide the number of invalid reports by the total number of reports to get the percentage of Noise.

This table shows the Signal percentage divided into four categories (red, yellow, green, gold), signifying underperformance to exceptional performance percentages, with suggested actions.


Signal percentage with suggested actions

Low Signal can be an indicator of many different things by itself, or a combination of elements.

> **"If a program is underperforming, we need to look at elements such as the program Scope, Policy, Staff, Rewards, Hackers, and Process. Some surface issues may be caused by other underlying issues, so it is important to perform a complete analysis of the entire program, from scope to process, to identify all areas for improvement before attempting to fix them. Fixing some issues without understanding the state of the whole program may result in causing or worsening other issues. "**
>
> **Chris Holt, Strategic Engagement and Community Architect, Intigriti**

## What happens if the scope is too small or too large?

If the scope is too small, you limit researchers; if it is too large, there is not enough focus on the important issues.

'Clearly define the scope of testing, include up-to-date documentation, known limitations, and provide test credentials where applicable. Security researchers sometimes submit invalid issues simply because they don't have a clear idea of the company's priorities. A strong scope not only reduces noise but also attracts quality researchers by helping them understand exactly what systems are in scope and what are not, meaning no time is wasted and no frustration is experienced in trying to understand what is allowed.' – How to attract security researchers to test on my bug bounty program

Program managers should consider making use of advanced scope definition tools and features in Intigriti's Asset Management tool suite, such as 'Asset Groups', 'Required Skills for Assets', 'Bounty Tiers', and 'Bulk Asset Import'.

## What happens if the scope is broad, but there are too many exclusions?

Programs with extensive exclusions make it harder for researchers to find bugs that qualify within the scope, which means more time and effort are invested and then lost. When comparing, researchers will

usually choose to work on programs that have a clear and broad scope with few limitations to halt or get in the way of their efforts, especially if the bounty reward is a comparable price. In short, simple categories optimize program engagement.

'Adding helpful guidance on how your tech stack works, including documentation and detailed FAQs, as well as adding information to the program on areas or vulnerabilities you particularly want to focus on, is a good way to ensure a researcher's time is spent testing the things you want them to.' – Nurturing program engagement

## What happens if a non-standard asset definition or structure is used?

If a non-standardized method is used to evaluate/describe assets and features, it makes it harder for researchers to document and report. Not only can it be unclear what non-standardised assets are included in the scope, but when assets are not logged or accurately tracked, this can limit the effectiveness of a program.

'When assets are not logged or accurately tracked, they fall outside the defined scope of the program, meaning researchers can't test them. This not only limits the effectiveness of bug bounty programs but also introduces compliance risks, as unmonitored assets may fail to meet regulatory standards. Since patching priorities often rely on CMDB data, critical vulnerabilities in overlooked systems may remain unpatched, undermining both the program's value and weakening the organization's overall security posture.' – Security maturity

## What happens if products are difficult to approach?

If products are highly complex, there are multiple restrictive rules in place, or are difficult to access, this can lead to reduced participation where researchers choose to hone their efforts on clear and accessible programs/targets instead. If a researcher can't buy, ship, probe, or monitor the product, then it might not be worth their time.

Ambiguous documentation also means confusion as to what is in and out of scope. Time could be spent testing an area out of scope, and then no bounty would be awarded for any findings; one of the most demoralizing moments for a security researcher.

What are potential issues when it comes to policy?

- **Lack of detail**: If the policy does not provide enough detail for researchers to understand restrictions, this can cause issues.
- **Professionalism**: Typos, grammatical errors, or unprofessional language.
- **Language**: If non-standard language is used for one scope and not in others, this can be confusing.
- **Structure:** If it is too long or uses many special cases to exclude from the scope, then it might not be worth the researcher's energy for the reward being presented.

- **IP Ownership:** Who owns the intellectual property described within a vulnerability report? Programs should make a commitment to respect IP ownership, and most researchers greatly appreciate retaining ownership.

- **Safe Harbour**: Researchers need to feel safe from legal threats. Safe Harbor is a baseline standard requirement and has multiple flavors to choose from based on the company's legal advisor's guidance.

As an example of this Safe Harbor commitment, Intel, when setting up a big bounty program with Intigriti, stated that 'If you follow the program terms, we will not initiate a lawsuit or law enforcement investigation against you in response to your report. Please understand that this waiver does not apply to your security research that involves the networks, systems, information, applications, devices, products, or services of another party (which is not Intel). We cannot and do not authorize security research in the name of other entities.'

## What are potential issues when it comes to interactions between researchers and company employees?

The key issue here lies in communication:

1. **Human interaction**: If employees are too robotic in nature, and there is not enough human interaction, then this can have a negative impact. Automation is very helpful in keeping status updates moving quickly. Remember, though, that this is a program built to work with people. Reflect on your personal experience when you go to a company for support or help, and all you get is a robot. If researchers are treated like screen names and not with the respect they deserve, this can impact working relationships.

2. **Skill issues**: Nobody expects program staff to be the all-knowledgeable expert on every product and vulnerability, and attack type. It is OK to say, "I don't know, let me talk to my team and get back to you." When staff do not understand the perceived risk being presented to them, or do not understand technical language, then that can have a negative impact on communication. It may be down to a lack of knowledge that results in staff rejecting too many reports, which, again, will have a negative impact.

'To maximize the signal-to-noise ratio of your bug bounty program, strive to build and maintain relationships with these expert hackers by showing your appreciation to them. This doesn't have to be through financial incentives, although you should always aim to reward every valid submission fairly and on time. Other ways to show appreciation include being professional in your correspondence, triaging issues quickly, and keeping researchers updated on the progress of the fix. This will build trust and attract top performers to your program.'- How to get valuable bug bounty reports

## How to reflect on your program reward structure

Your program policy is your commitment to researchers. It's a promise that when they abide by your rules and provide value within your prescribed bounds, you will respond with gratitude and the appropriate rewards. Rewarding the right amount of bounty is critical.

**Too low:** Bounties can be too low for the time required to hunt for vulnerabilities. Researchers favour programs that reward the difficulty level and time required to find meaningful vulnerabilities.

**Too high:** If bounties are high but the program has never paid top-level rewards, this shows an issue with credibility and may reduce the level of trust researchers have in the program/company. Think about your highest-paid reports and why they did not reach the maximum level of payment.

*Program owners considering an adjustment, ask your Customer Success Manager or Accounts team for advice, and spend some time planning with Intigriti's Bounty Calculator tool.*

**Response time**: Even if the payment amount is correct, the time between submission and payout can be too long, which can deter researchers equally. Look to your internal process for how reports are processed; perhaps efficiency can be gained by introducing automation or removing/simplifying some decision tasks.

**Custom Severity Scoring**: The two most common types of severity scoring systems are CVSS (v3.1 or v4.0) and a basic 6-point scale (None, Low, Medium, High, Critical, Exception). Custom severity scoring systems can work, but should be transparently documented so that researchers can predict their score and even provide the prediction in the report. If you are using CVSS, consider enabling Ranged Bounties in your Intigriti program for more granular payout results.

**Strict nondisclosure requirements**: These deter talented researchers from participating, as companies may delay patching without public pressure. It also prevents community-wide learning. Some researchers just want to be paid, but many others value the ability to blog about their work, earn named recognition in CVE listings, and showcase their work at conferences or community events.

## Why does having the right researchers matter?

There are many elements to consider when aligning the right researchers with the right program. Elements including:

**Language**: Language barriers and skill levels can impact how a person interprets product or usage structures and security models. Researchers are a global community, ranging from novice to expert, child to adult. The language you speak and read might not be the researcher's first or even fluent language. This is why it is important to write using a standardised format, and to avoid abbreviations, acronyms, and jargon.

**Skillset**: If the skillset does not align with the company's interest in product requirements, this can mean that the wrong hackers are engaged. Not every hacker is a fit for every program. Intigriti has introduced platform features that define and track skills and interests to help identify researcher-to-program and researcher-to-asset matches.

**Restrictions in place:** Restrictions on researchers, such as requirements for social security numbers, VPNs, or IP whitelisting, need to align with what the company and researcher feel comfortable with. Researchers may not be comfortable with the requirements in place from a company or program. If you have strong testing restrictions in place, consider whether they come from a place of technical necessity, or fear of risk and distrust; trust can be built.

# Next steps to enhance your bug bounty journey

To answer any questions on the topics raised in this article, contact the team today.

Or, to further optimize and strengthen your bug bounty program to maximize security impact, these points can help you build, structure, and improve for researcher-friendly, modern, and effective programs.

- How to attract security researchers to test on my bug bounty program?
- What is the pattern that can be expected after going public with a bug bounty program?
- How should I scope third-party assets in my bug bounty program?
- How can I get more bug bounty submissions and higher-severity findings?
- How do I know I'm paying the right amount of bug bounty?
- Layered security in action: How VDP, Bug Bounty, and PTaaS combine to protect your business
- At the forefront of ethical hacking: What's Intigriti's impact and position?

AUTHOR

## Eleanor Barlow

Eleanor Barlow is a London-based Senior Cyber Security Technical Writer at Intigriti, with 9+ years' experience reporting on and writing for the cyber and tech sector. She specializes in data-driven content on cybersecurity and bug bounty intelligence, helping organizations benefit from the latest trends and insights.