| TECHNICAL BRIEF

tenable™
network security

# Detection Algorithms in Log Analysis

## Sifting the Needles from the Haystack

As the volume of log data generated in networks continues to grow, security practitioners have the challenge of detecting problems and anomalies quickly enough to take action and mitigate damage. To do this, they must constantly tune and refine detection algorithms.

Tenable's Chief Security Officer, Marcus Ranum, and fellow security practitioner, Ron Dilley, have dedicated much of their careers to finding better ways to detect anomalies and threats within log data. As part of an ongoing Tenable webinar series on log analysis topics, the pair shared their insights and experiences. This paper summarizes some of the key points and recommendations from their discussion, available here.

## Key Points

No matter how advanced your detection algorithms, they fall into two main strategies: pattern matching and statistics/probability. Even the most sophisticated algorithms create some variation of a whitelist, blacklist or graylist, or determine whether an event is improbable (and hence should be gray-listed).

**Finding the black swan:** Pattern matching has an inherent limitation – you can only match the patterns that you already know about. How do you find the never-before-seen pattern (the 'black swan') in your data, if you do not yet have the pattern?

One approach is to define a baseline of normal behavior, then identify any deviations from that baseline. When you discard what is normal, what's left is inherently interesting to the security practitioner, particularly during incident response.

**Signal-to-noise ratio:** Because you need a timely answer to your questions, you need to be able to limit the data set. As an analyst, rather than focusing solely on improving pattern matching, it's important to spend time improving the signal-to-noise ratio. By eliminating the normal or irrelevant, you can focus on what's unusual and get answers quickly enough to take action.

> "You can build a tool that finds a needle in a haystack, or build a tool that removes the hay and scrutinizes what's left." — ***Ron Dilley***

Finding ways to optimize the signal-to-noise ratio should be part of the up-front development time in any log analysis system or project. You need to set aside time to have discussions about what constitutes normal activity, and how you can partition your data for analysis in a way that will help you find anomalies. This is true whether you are using packaged solutions or writing your own code.

> "Whether you're using your own code or off-the-shelf systems, you have to spend time writing filters and rules for it, if you want to have success."
>
> — ***Marcus Ranum***

## Recommendations

**Start by logging everything:** The first place to start is by logging nearly everything – system logs, web server logs, etc. For DNS systems where logging may slow DNS responses, you can attach a custom DNS sniffer on a tap or mirrored port that understands DNS traffic and logs it.

Bandwidth is rarely the limiting factor, but processing time can be, particularly when it comes to incident response. If you do a good job of filtering and removing the 'noise' from log data, then you can make better use of machine memory for analysis and incident response.

**Use front-end aggregators:** For large networks, consider using dedicated front-end aggregators to preprocess data, using the same processing algorithms across all systems. This creates a very scalable model for log processing and analysis. Then, combine the results in a central repository for future analysis. Even at this stage, there are different approaches for managing the data:

- Preprocess the data to sift out the 'noise', send the summaries only to the aggregation point, and discard the log data or retain it at the edge.
- Preprocess the data at the front end, then add the summary information and send the enriched data to the central aggregation server.

The danger with discarding the origin data is that you may need to retroactively analyze it when you discover a new pattern or problem. For this reason, it's best to retain all of the original log data.

> "I love the time machine effect of having two years of logs online. You can go back and answer questions about what happened three months ago." — ***Ron Dilley***

**Create a network flight recorder:** For more incident response insight, use a network recorder (packet vacuum) to store every packet on your network for 15-30 days. You can set this up with storage to work as a flight recorder – discarding the oldest data when the disk fills up and continuously cycling through storage. This data can be invaluable when troubleshooting.

**Reduce the noise:** Use multiple filters and strategies to emphasize the significant and de-emphasize the irrelevant. Find ways to partition your data for analysis. For example, if you know that all of your web data is in a specific directory, filter change detection for those files differently than files in other directories.

> "You know and understand your traffic and can use that understanding to partition your efforts– for example, doing log analysis on traffic to/from HR machines, or changes in website content. By partitioning the data, you can more easily disregard the noise."
>
> — ***Marcus Ranum***

**Find anomalies:** Use multiple strategies to look for many small indicators of anomalous activities.

- Use tools like tarpits, honeypots , false files or URLs that never have a legitimate reason for showing up in monitoring tools. These instruments have an inherently good signal-to-noise ratio.
- Look at the shape of data rather than its content. For example, a 50K syslog message is an anomaly that merits inspection.
- Identify normal event chains, such as known URL sequences in normal web traffic. Then look for patterns that do not match.
- Fingerprint browsers using tools like PChat to find known good browser configurations – and alert on any other browsers.
- Identify what happens when a workstation boots up, including which calls are made in which order. When a workstation boots and does not match this pattern, something may be wrong.
- Look for error rates. When a machine attempts to do something that is not permitted, that can raise a red flag. Or if someone doesn't know your way around your network and/or what processes you have turned off, they will leave traces in the error messages they generate.

> "In terms of signal-to-noise ratio, things that are permitted are less interesting than things that are not permitted." — ***Ron Dilley***

Taking it further, you can use a dynamic 'risk scoring' strategy, similar to the risk scoring used in fraud prevention. These systems can be tremendously powerful depending on how you set them up. For example, you might increment the risk score when a system sends a DNS query, then decrement the same amount when they get a response. As part of normal operations, DNS behavior should resolve to zero risk. A botnet participant looking for the new Command and Control system will search for domains based on an algorithm – generating a very high DNS-based risk score.

To listen to the original webcast, visit tenable.com.

## About Tenable

Tenable Network Security is relied upon by more than 15,000 organizations, including the entire U.S. Department of Defense and many of the world's largest companies and governments, to stay ahead of emerging vulnerabilities, threats and compliance-related risks. Its Nessus and SecurityCenter solutions continue to set the standard for identifying vulnerabilities, preventing attacks and complying with a multitude of regulatory requirements. For more information, please visit www.tenable.com.

## For More Information

**Questions, purchasing, or evaluation:**
subscriptions@tenable.com or 410.872.0555, x506
Twitter: @TenableSecurity
YouTube: youtube.com/tenablesecurity
Tenable Blog: blog.tenable.com
Tenable Discussions: discussions.nessus.org
www.tenable.com

tenable™
network security