tenable network security

Security Data Aggregation:
Modeling the Security 'Big Data' Challenge

IT security teams collect more data every year, from sources across the network, with the goal of obtaining better telemetry and visibility. This creates a 'big data' challenge when it comes to security – how do you collect, aggregate and work with that data in a way that helps you solve complex security problems?

As part of a new webinar series on security topics, *Geeking Out with Marcus Ranum*, Tenable hosted a webcast with guest speaker, Ron Dilley, on the topic of Security Data Aggregation. The topics covered include what data to aggregate, how to aggregate it, and how to derive insights from the data. This paper summarizes the key points and recommendations from the discussion.

Key Points

When it comes to security incident response, having more data is almost always better. Ron suggests that organizations log as much as possible, as it is difficult to anticipate in advance the type of information you might need during an incident.

Logging has value from both security and operational perspectives:

- For security teams, log event data helps you discover what bad actors did once past your security controls. Logging helps you detect and determine the extent of a breach so you can react more quickly.
- For operations teams and administrators, log event data provides invaluable information about what is going on when a system or application experiences problems. Security administrators can sell the operational value of logging to get buy-in from individual administrators who may resist turning on logging on their systems.

When people object to logging, they usually worry about bandwidth, storage costs, and performance impact. In most cases, these concerns are based on misconceptions.

- Bandwidth concerns are overrated; even 400 - 500G of data/day translates into a small amount of bandwidth consumption per second at the point of aggregation. Even systems that generate large amounts of logs during the course of the day consume a relatively small amount of data per second to send those logs to an aggregation point. If you don't have the spare bandwidth for logging, you're probably running into other bandwidth problems.
- Storage is cheap, relatively. With open source software and commodity storage, you can create highly scalable storage environments at a relatively low cost. (See the recommendations for storage architectures below.)
- Performance concerns are usually overrated. In some cases, however, you may need to create workarounds such as setting up passive sniffers to get specific packets near a particularly delicate system. If blocking I/O in TCP presents a problem, you can get around it by using UDP.

Resistance to logging is usually based on misconceptions. From a management perspective, logs provide invaluable insight in the case of a data breach or other security incident. However, it's always a danger to lead with fear – many times people cannot see a fear-based justification until they have experienced the loss. Instead, you can make a compelling cost-based argument. When you do not have logs, any security incident response will be orders of magnitude more costly and time-consuming than it is with logs. Without logs, you may spend weeks or months determining the extent of a breach, while with logs you can be on the leading edge of the breach.

> "I've never come across a situation where having logging hurt anything or caused outages. But I cannot count the number of times that having the data has saved my bacon."
>
> ***Ron Dilley***
> A security practitioner with expertise building enterprise security programs and architecture.

Recommendations

Log data provides invaluable insight for security and operational purposes alike. The challenge is in aggregating the large volumes of data from multiple sources and normalizing and analyzing it effectively. As someone who has dedicated a good deal of his career to this very effort, Ron Dilley provides valuable insight and suggestions.

Log everything: Ron's approach in any new situation is to start by logging everything – at all layers. It's easier to start with everything and selectively turn off streams than it is to figure out all the things you might need. Then find the noisy logs and turn them off or adjust the logging settings.

- In UNIX – Auditd gives you visibility into what processes are executing and what files are being touched
- Use the *.debug setting in your UNIX syslog configuration file to catch all logging levels in syslogs
- On Windows – log everything and convert Windows logs to the syslog format.
- Web logs – use software like ModSecurity to track what's happening
- DNS and DHCP - Being able to map a DNS name to an IP address in the past is very valuable, so there's a strong case for keeping DNS records. DHCP servers can generate a lot of noise with lease renewals, and may be one of the systems that you want to turn off logging for once you start pruning. There is a tool available that gathers MAC address, current IP address and login ID whenever someone logs into a system – eliminating the need for DHCP data.

How long you retain logs depends on many factors, including your organization's tolerance for risk and the sensitivity of the legal department. Based on experience, two years is a good amount of time to store data to address security incidents, operational problems, and staff investigations.

> "I'm a big believer in keeping data for two years. Invariably you have to go back more than one year to find data that you need." – ***Ron Dilley***

Data models: Ron prefers to store log data in raw ASCII files and use post-processing to normalize and analyze it. However, because speed of analysis can be critical during an incident, you might want to set up a hybrid data model using Hadoop or some other technology for faster access to recent logs.

Storing logs: You can create cost-effective, scalable log storage using clustered file systems and open source storage software, such as Red Hat Gluster. This approach lets you use commodity storage for performance and redundancy. Then compress the logs as you store them to reduce your storage costs. Using parallel compression works well for high data volumes.

Analyzing logs: Once you have aggregated the log data, you can start analyzing for problems and patterns.

- Start by looking for known bad actors (black and grey lists)
- Use signature-based analysis to find known threats or problems.
- Look for patterns and abnormalities in the data. With enough data, you can identify a baseline of what's normal, then trigger alerts when anomalies appear.

Look for anomalies: Attackers are good at evading signature detection and poorly configured security controls, but malware inevitably leaves traces. "Low and slow" attacks are designed to hide from pattern-based intrusion detection systems, but they generate their own patterns which you can detect if you are looking for anomalies. In the case of a low and slow attack, the pattern might be infrequent, short-duration connections made from the same IP address. By looking at patterns over time you can often detect these exploits.

> "Spend less time looking for the boat and more time looking for its wake." — ***Ron Dilley***

Looking for simple patterns like bytes in/bytes out can be very effective. If normal traffic on a web application has a basic ratio of bytes in to bytes out and suddenly that ratio flips, it's an anomaly that needs investigation.

Vacuum the packets: For even more data, use a packet vacuum to store every packet on your network from 15 - 30 days.

Integrate vulnerability assessments: If you have vulnerability assessment information at your fingertips, you can make smarter and faster decisions. For example, you might detect a SQL injection attempt. If you have integrated vulnerability analysis with log analysis, you can know immediately whether or not the affected system is in fact vulnerable to that attack.

> "If you have integrated vulnerability analysis with log analysis, you can know immediately whether or not the affected system is in fact vulnerable to that attack."

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