« Previous: AppSec Decoded: Why Biden's… (https://www.synopsys.com/blogs/software-security/appsec-decoded-executive-order/)
Next: Software risks in private equity… (https://www.synopsys.com/blogs/software-security/software-risks-private-equity-buyouts/) »

Top seven logging and monitoring best practices

Posted by Ashutosh Rana (https://www.synopsys.com/blogs/software-security/author/ashutor/) on Monday, November 1, 2021

Need to build a logging and monitoring solution and unsure where to begin? Get started with our logging and monitoring best practices guide.

By: Nivedita Murthy (https://www.synopsys.com/blogs/software-security/author/nmurthy/), senior security consultant, and Ashutosh Rana (https://www.synopsys.com/blogs/software-security/author/ashutor/), senior security consultant, at Synopsys.

The concept of logging and monitoring isn't new, but organizations still struggle to formulate and implement a security-focused logging and monitoring policy. Security teams need to build logging and monitoring programs that not only collect traditional operational metrics, but are also capable of storing, analyzing, and even mitigating a variety of attacks. This proactive approach of collecting and analyzing information can help developers, sysadmins and security teams in many ways such as detecting issues in their code via "application-level logging," identifying anomalies in network traffic via "infrastructure logs like AWS/Azure," and detect as well as prevent security incidents by using advance Security Information and Event Management capabilities. You can address these challenges by employing these logging and monitoring best practices.

devsecops.html?intcmp=sig-blog-os-rightrail)

RELATED TAGS

Application sec
(https://www.sy
security/tag/ap
practices/)

SEE ALL TAGS

Explore the Synopsys full course catalog (https://www.syr

1. Define your need to log and monitor

Determining why the organization wants a logging solution will help define what you need to log. The following are some of the reasons an organization might want such as solution:

- Compliance requirements
- Local laws and regulations
- Incident response requirements

Discussing these factors with your organization's security governance team, legal department, and other stakeholders will help define the goals for logging and monitoring.

2. List what needs to be logged and how it needs to be monitored

Based on your goals, determine what metadata needs to be captured and what events need to be logged. Some examples of metadata and events to be logged and why include:

- PII/PHI transactions to be HIPAA compliant (https://www.synopsys.com/glossary/what-is-hipaa.html)
- Financial transactions to be PCI DSS complaint (https://www.synopsys.com/glossary/what-is-pci-dss-

compliance.html)
- Authentication attempts to a server (successful and failed logins, password changes)
- Commands executed on a server
- Queries (especially DML queries) executed on a database

Infrastructure administrators and security teams (https://www.synopsys.com/software-integrity/solutions/security-teams.html) should collaborate to build an effective logging and monitoring program that collects traditional operational metrics and can analyze them to mitigate attacks. Alerts on certain events, such as multiple failed login attempts or weekly notifications on commands executed on a server, can be set up to monitor these events. It is also important to work with application teams (https://www.synopsys.com/software-integrity/solutions/dev-devops.html) to understand what the different attributes of a log entry mean. Once you have a baseline for normal operations, you can configure correlation rules, aggregations, thresholds, and alerts to be triggered for any anomaly based on the security risk profile for the application. For example, every log entry should have at least the following:

- An actor (who: username, IP address)
- An action (what: read/write on which resource)
- A time (when: timestamp)
- A location (where: geolocation, browser, code script name)

3. Identify assets and events that need to be monitored

Log data is a huge volume of datasets that impact performance and costs. When determining what data you should monitor start by not leaving anything out. You need to identify which systems/applications should be monitored and what level of monitoring is required. You should also classify your data and systems according to the organization's statutory, regulatory, or contractual requirements. Keep in mind that this classification may differ from your security system classification or your business data classification.

4. Determine the right solution for logging and monitoring

There are a many solutions—both commercial products and open source projects—to choose from when you want to build a scalable and resilient logging and monitoring program. Choosing the right technologies for a logging and monitoring architecture can be overwhelming. A few key points that you need to keep in mind are:

- Automate as much of the monitoring process as possible
- Constantly tune your alerts and log sources as threats evolve
- Ensure that log and alerts are generated in a standardized format

5. Design logging and monitoring systems with security in mind

A logging and monitoring program by itself is an asset to the organization because it looks into organization wide activities and may contain sensitive information. Here are few points to consider to secure it:

- Redact/mask/anonymize sensitive information from event logs beforehand, to prevent sensitive information from being logged in plain text (e.g., PHI/PII information)
- Enforce role-based access controls
- Perform log integrity checks to ensure that logs are not tampered with
- Apply encryption at rest and transit
- Follow the principle of least privilege when configuring log sources
- Sanitize logs before storing and processing
- Include capabilities for high availability and redundancy

6. Adopt organizationwide logging and monitoring policies

Work with security teams to enforce companywide policies and procedures that define logging requirements in detail for all systems. This ensures consistency and that protocols and procedures are followed in logging. Policies with a strong mandate and corporate backing ensure that logging and monitoring practices are followed.

7. Establish active monitoring, alerting and incident response plan

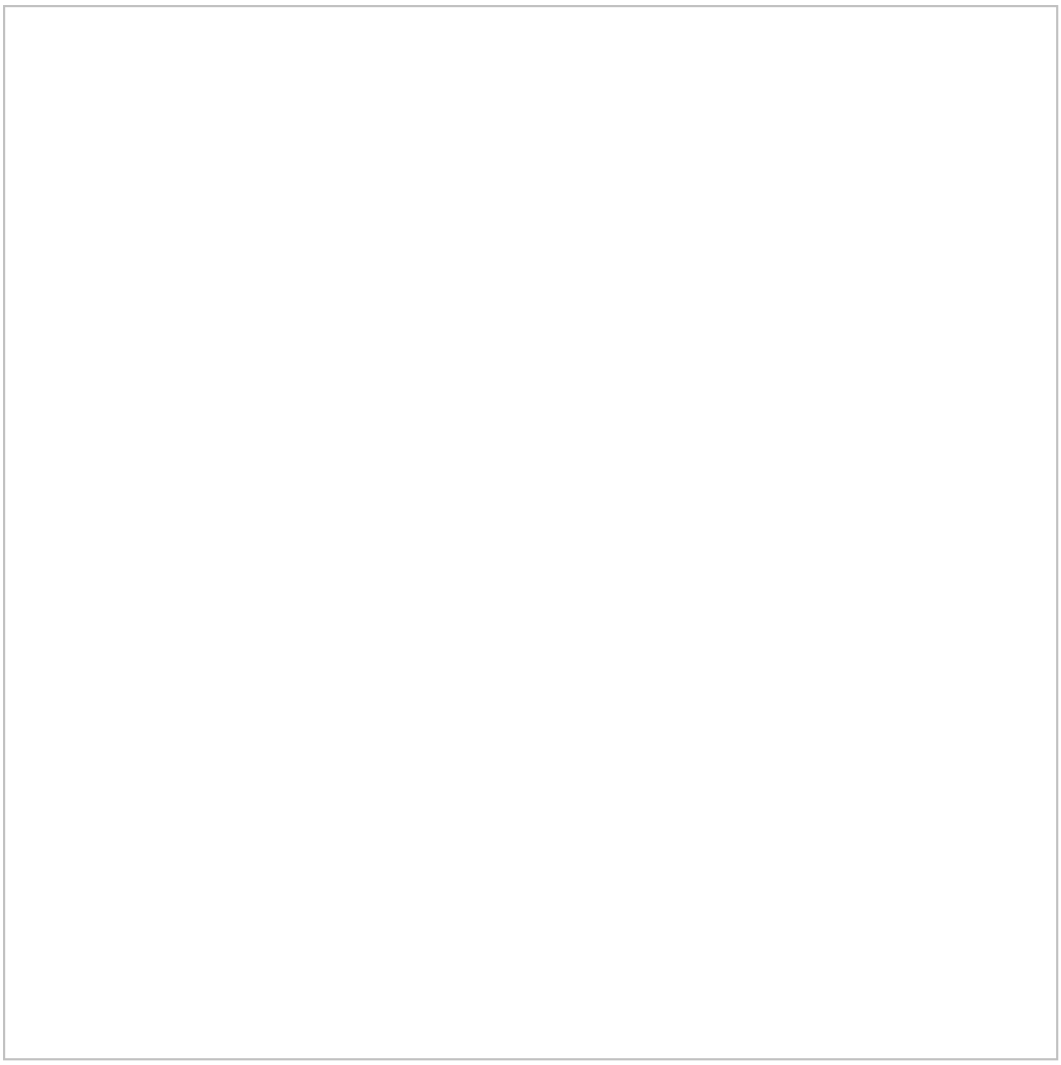
Without strong logging mechanisms, an organization is truly in the dark before, during, and after any incident. Attacks on sophisticated systems are often carried out for months or even years. Would your organization be able to detect and block a probe like this? If a motivated adversary can slowly pick apart an application for that length of time and go undetected, there is a high chance that an actual exploit will occur. The following steps are vital to prevent such a scenario:

- Establish an incident response plan and rehearse it at regular intervals
- Trigger alerts in an adequate amount of time
- Take active automated actions on the alerts

Although it is no easy task to build a secure logging and monitoring program, it is an imperative part of any application architecture and it will make all the difference in detecting and blocking a sophisticated attack by a motivated and determined adversary. Explore Synopsys eLearning courses to learn more on logging and monitoring best practices.

Application security best practices course catalog | Synopsys

(https://www.synopsys.com/software-integrity/training/software-security-courses.html)



This post is filed under Building secure software (https://www.synopsys.com/blogs/software-security/category/secure-software-development/).

Ashutosh Rana

Posted by

Ashutosh Rana

Ashutosh Rana is a senior security consultant with over 10 ye…

SEE AUTHOR ARCHIVE (https://www.synopsys.com/blogs/software-security/author/ashutor)

More from Building secure software

OWASP API Security Top 10: Security risks that should be on your radar (https://www.synopsys.com/blogs/software-security/owasp-api-security-top-10/)

AppSec Decoded: Security at the speed of DevOps (https://www.synopsys.com/blogs/software-security/appsec-decoded-security-at-speed-of-devops/)

Posted by
Charlotte Freeman (https://www.synopsys.com/blogs/sof security/author/cfreeman/)
on July 5, 2022

API security testing (https://www.synopsys.com/bl security/tag/api-security-testin

Software compliance, quality, a (https://www.synopsys.com/bl security/tag/software-quality-c

Posted by
Synopsys Editorial Team (https://www.synopsys.com/blogs/sof security/author/synedt/)
on June 10, 2022

Application security orchestrat correlation (https://www.synopsys.com/bl security/tag/appsec-orchestra correlation/)

CyRC Case Study: Securing BIND 9 (https://www.syno security/cyrc-case-study-securing-bind-9/)

Posted by
Jonathan Knudsen (https://www.synopsys.com/blogs/sof security/author/jknudsen/)
on May 24, 2022

Cybersecurity Research Center (https://www.synopsys.com/bl security/tag/cybersecurity-rese

Fuzz testing (https://www.synopsys.com/bl security/tag/fuzz-testing/)

Building security into existing source code management workflows (https://www.syno security/building-security-into-existing-scm-workflows-code-dx/)

Posted by
James Rabon (https://www.synopsys.com/blogs/sof security/author/rabon/)
on May 17, 2022

Application security orchestrat correlation (https://www.synopsys.com/bl security/tag/appsec-orchestra correlation/)

Static application security testi
(https://www.synopsys.com/bl
security/tag/static-analysis-sa

DevSecOps
(https://www.synopsys.com/bl
security/tag/devsecops/)

Software Integrity Group's prod
services
(https://www.synopsys.com/bl
security/tag/appsec-product-o

synopsys.html) (https://www.