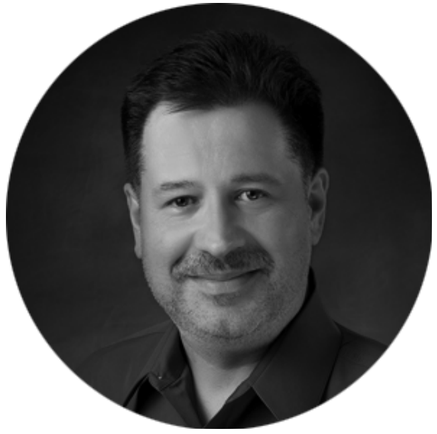
Security Logging and Monitoring Failures



Alexander Tushinsky

Cybersecurity & Software Development Consultant

@ltmodcs alextushinsky.com

Overview

Security Logging and Monitoring Failures

- What should we log and what events should we track?
- Logging and notification in .NET.

Logging & Monitoring

- Identification of security incidents
- Monitoring for policy abuse
- Non-repudiation and traceability
- Detection of problems
- Audit, compliance, and incident investigation

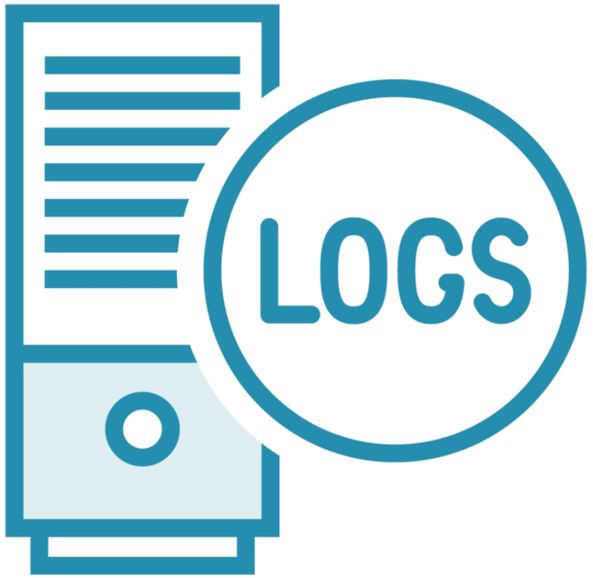
Security Logging and Monitoring Failures

- Not a vulnerability
- Logs may reveal sensitive data
- Logs don’t provide enough detail to identify an issue

Logging – Why?

- Events occur at the application level (target, action, outcome)
- User details (identity, roles, permissions)

Logging – Where?



- **File System**
  - Segregate logs
  - Should not be web accessible
- **Database**
  - Separate database with restrictive permissions
- **Cloud**
  - Azure Insights

Logging – What?

- Input validation failures
- Authentication success and failures
- Authorization failures
- Session failures
- Application exceptions
- High-risk functionality
- Legal requirements

Logging – Log Detail

- Date/time
- Interaction identifier
- Application identifier
- Entry point
- Username, IP address
- Type of event
- Severity of the event
- Description

Logging – What Not to Log

- Source code
- Session / Access identifiers
- Sensitive personal data
- Passwords
- Connection Strings
- Encryption keys
- Information not legally trackable

Logging – Monitoring

- Connect to SIEM
- Enable SMS or email alerts, if necessary

Demo

Azure Insights Integration

- Default logging in .NET 6
- Add Azure Insights to an application
- Create a log entry
- Review output

Summary

Security Logging and Monitoring Failures

- Reviewed the why, what, and where of logging
- Looked at the default logging in .NET 6
- Integrated with Azure Insights

Up Next:
Server-Side Request Forgery (SSRF)