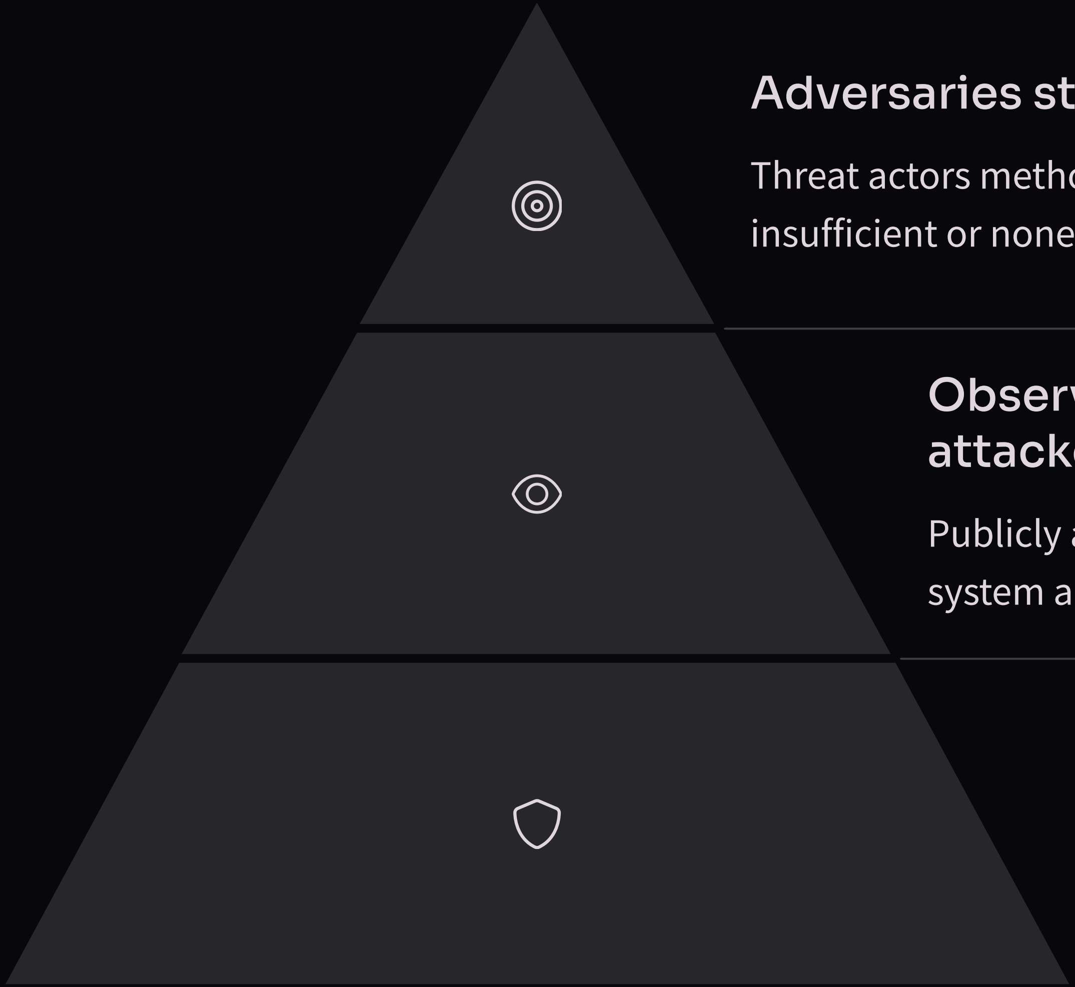
Threat Modeling for Observability: From Blind Spots to Actionable Insight

A practical guide for security engineers and DevOps professionals to identify, prioritize, and mitigate security risks in observability systems.

Understanding the Observability Blindside



Adversaries strategically target monitoring gaps

Threat actors methodically exploit areas where observability coverage is insufficient or nonexistent

Observability data becomes intelligence for attackers

Publicly accessible metrics, logs, and traces inadvertently reveal system architecture and potential entry points

Vigilance requires continuous reassessment

Even robust observability implementations harbor hidden vulnerabilities that evolve with your system

The Traditional Observability Triad

Metrics

Numerical representations of system behavior over time.

- Performance indicators
- Resource utilization
- Error rates

Logs

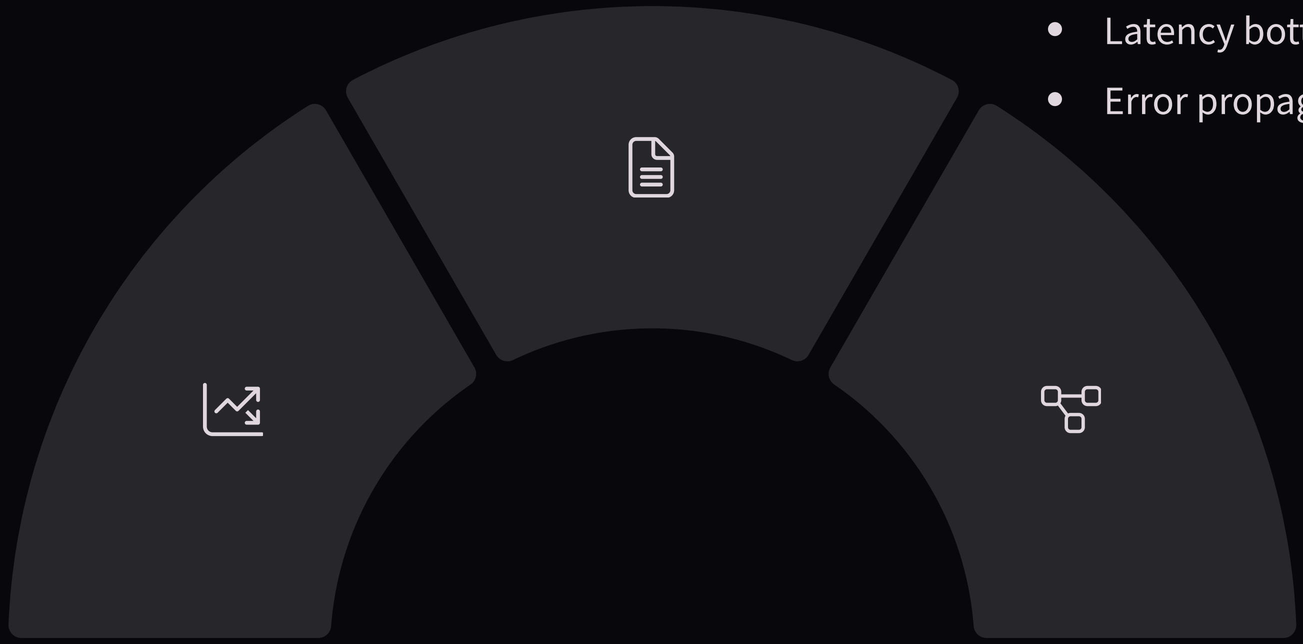
Timestamped records of discrete events within systems.

- Error messages
- Access attempts
- State changes

Traces

End-to-end request flows across distributed systems.

- Service dependencies
- Latency bottlenecks
- Error propagation



Security Risks in Observability Pipelines

Collection Points

Unsecured agents and collectors expose entry points.

- Outdated agent software
- Excessive privileges
- Unencrypted transport

Transport Layer

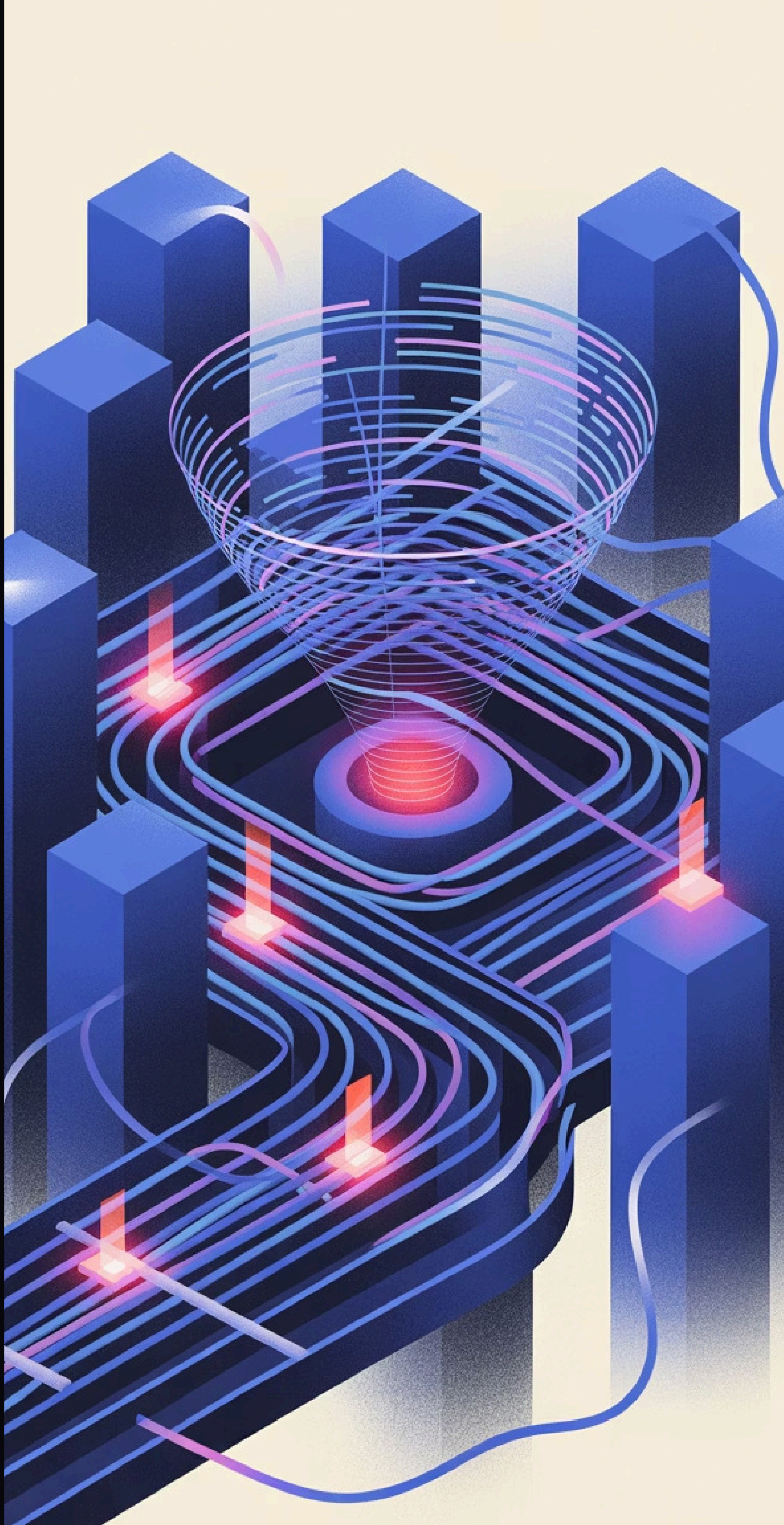
Data in transit becomes vulnerable to interception.

- Missing TLS encryption
- Certificate mismanagement
- Weak cipher suites

Storage Systems

Centralized telemetry creates high-value targets.

- Insufficient access controls
- Unpatched vulnerabilities
- Improper data retention



Threat Actor Motivations

Reconnaissance

Attackers leverage exposed metrics to map your infrastructure. They identify server locations, software versions, and potential entry points through misconfigured dashboards.

Data Exfiltration

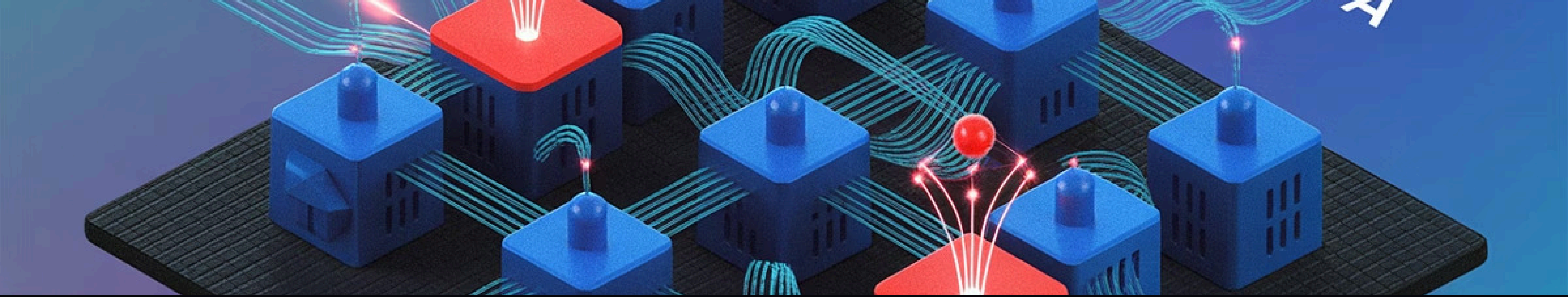
Sensitive information embedded in logs becomes a target. PII, secrets, and access tokens accidentally logged create compliance violations and security breaches.

Alert Fatigue

Threat actors deliberately trigger false positives. Overwhelming alert systems creates operational blindness to actual security incidents.

Living Off The Land

Attackers use legitimate observability tools. Monitoring systems with elevated privileges become perfect vehicles for lateral movement.



Mapping Attack Surfaces

Identify Assets

Document all observability components in your architecture. Map data flows between collectors, processors, and visualization tools.

Define Trust Boundaries

Establish where data crosses security domains. Determine which components have privileged access to sensitive systems.

Enumerate Entry Points

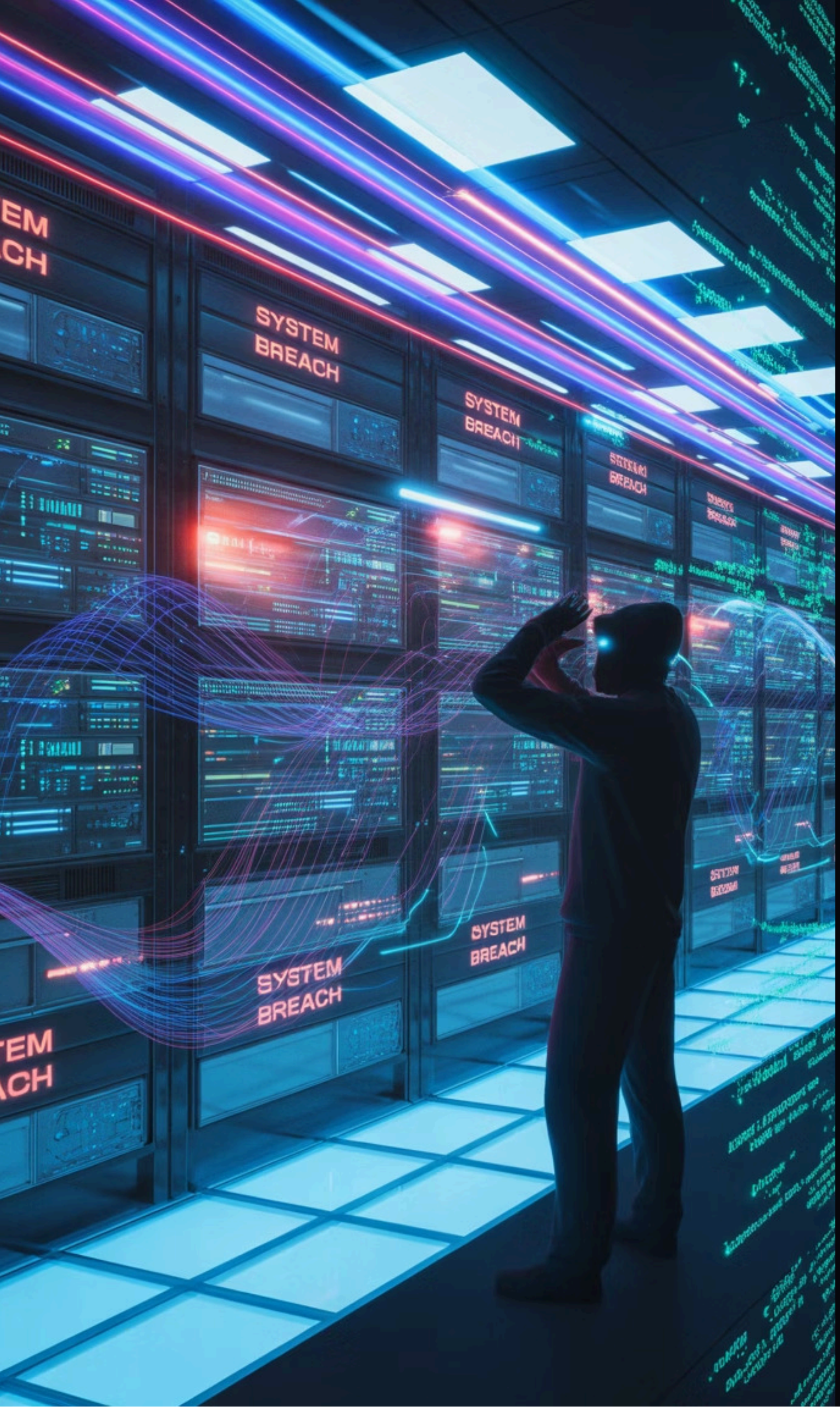
Catalog all interfaces exposed by monitoring tools. Consider API endpoints, dashboards, and agent communication channels.

Prioritize Threats

Rank vulnerabilities by potential impact and likelihood. Focus remediation efforts on critical observability components first.

STRIDE for Observability Systems

Threat Type	Observability Impact	Mitigation Strategy
Spoofing	False metric injection	Strong authentication for all agents
Tampering	Modified telemetry data	Cryptographic integrity checks
Repudiation	Deleted audit logs	Immutable logging pipelines
Information Disclosure	Exposed sensitive metrics	Strict access controls on dashboards
Denial of Service	Overloaded collectors	Rate limiting and redundancy
Elevation of Privilege	Compromised monitoring agents	Least privilege principle



Real-World Attack Scenario

Initial Access

Attacker exploits outdated Grafana instance. Public dashboard exposes internal infrastructure details.

Credential Theft

API keys extracted from plaintext logs. Monitoring service accounts have excessive permissions.

Lateral Movement

Prometheus server becomes pivot point. Attacker moves from monitoring to production infrastructure.

4

Data Exfiltration

Custom metric queries extract sensitive data. Exfiltration blends with normal monitoring traffic.

Securing Observability Pipelines

100%

Encrypted Telemetry

All monitoring data must use TLS in transit

2FA

Dashboard Access

Require multi-factor for all monitoring interfaces

30d

Rotation Schedule

Maximum lifetime for monitoring credentials

0

Secrets in Logs

Tolerable number of credentials in telemetry

Building Alert Resilience

Implement Signal-to-Noise Filtering

Develop correlation rules that reduce false positives. Group related alerts to prevent alert fatigue.

Establish Alert Tiers

Create severity-based routing for notifications. Critical security alerts must bypass throttling mechanisms.

Employ Anomaly Detection

Implement ML-based detection of unusual patterns. Baseline normal behavior before deploying alerting.

Protect Alert Mechanisms

Treat notification systems as critical infrastructure. Attackers often target alerting to hide activities.

Action Plan: From Blind Spots to Insight

Conduct Observability Threat Assessment

Map your current monitoring architecture

Implement Security Controls

Secure data at collection, transport, and storage

Integrate with Security Operations

Align monitoring and security teams

Measure Security Effectiveness

Track security metrics for continuous improvement

Transform your observability from a potential liability into a security asset. Start with a comprehensive assessment, then systematically address vulnerabilities.

Thank you