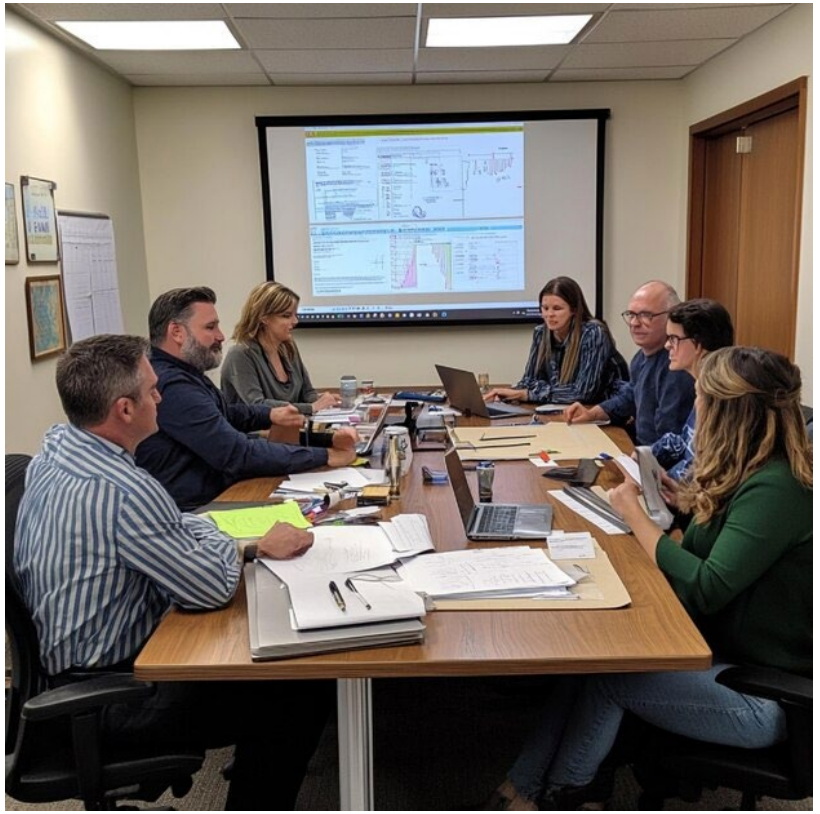
RISK AND RESILIENCE BOOTCAMP

TEKsystems Global Services

WORKFORCE DEVELOPMENT

RISK RESPONSE PLANNING

This module is an overview on developing a high maturity risk response plan



MITIGATION PLANNING

- Mitigation planning
  - How an organization turns abstract risk analysis into concrete, executed change
  - Risk assessment describes what could go wrong
  - Risk response chooses how to treat it (avoid/mitigate/transfer/accept)
- The mitigation plan answers
  - “Exactly what will we do, by when, with which resources, and how will we know we’ve succeeded?”
- In mature environments
  - Mitigation plans are tightly integrated with
    - Risk registers, GRC workflows, project management, and operational metrics (KPIs/KRIs)
    - Poorly defined mitigation is one of the most common failure modes in risk program
    - Risks get “accepted by default” because mitigation is vague, under-resourced, or untracked

MITIGATION PLANNING

- A robust mitigation plan has four pillars
  - Resources
  - Timelines
  - Success criteria
  - Clear ownership

RESOURCE PLANNING

- Mitigation without adequate resources is performative
  - It exists only on paper
  - Resource planning ensures the organization has the resources to implement the plan
- Technical resources
  - These are the people who actually change things in systems
    - Engineers / Developers: apply code changes, refactor applications, improve error handling
    - Platform / Cloud Engineers: adjust infrastructure, implement auto-scaling, reconfigure IAM
    - Security Analysts / Engineers: implement new controls, tuning SIEM/SOAR rules, deploy EDR
    - Architects: redesign critical parts of the system (e.g., remove single points of failure, break monoliths, segment networks)

RESOURCE PLANNING

- Technical resources
  - Example
  - Mitigation: “Reduce lateral movement risk in internal network”
  - Technical Resources
    - Network engineer to implement segmentation
    - Security engineer to tune microsegmentation policy
    - Architect to define zero-trust design patterns

RESOURCE PLANNING

- Financial resources
  - Mitigation often requires spend, including
    - New tools (e.g., CSPM platform, PAM solution, backup system)
    - Additional cloud capacity or redundancy
    - External consultants or penetration testers
    - Training programs for staff
  - Risk teams must coordinate with Finance and leadership to secure budget
  - Expenditures often justifying it with evidence for
    - Risk-reduction value
    - Cost of inaction vs. cost of mitigation
    - Regulatory/penalty avoidance

RESOURCE PLANNING

- Human resources: owners and SMEs
  - Mitigation needs named people, not just roles
    - Risk Owner: accountable for risk closure and residual risk
    - Mitigation Owner / Project Lead: coordinates the actual remediation activities
    - Subject Matter Experts (SMEs): cryptography, identity, DB, cloud, etc.
  - High-maturity organizations track workload to ensure people tasked with mitigation actually have capacity, not just theoretical assignment

RESOURCE PLANNING

- Infrastructure resources
  - Mitigation often requires safe testing and rollout environments, including
    - Non-production environments that resemble production (staging, pre-prod)
    - Sandbox environments for testing patches, configuration changes, and failover
  - Extra cloud capacity for
    - Blue/green deployments
    - Rolling updates
    - DR testing
    - Parallel runs

RESOURCE PLANNING

- Infrastructure resources
  - Example:
  - Mitigation: “Implement real-time database replication across regions”
  - Required infrastructure
    - Additional DB nodes in second region
    - Replication network setup
    - Load balancer configuration
  - Without infrastructure planning, mitigation may raise new risks
    - For example: untested failover, performance degradation

TIMELINES

- Timelines transform mitigation
  - From “we should do this”
  - Into “we will do this by X, and we will measure progress along the way”
- Risk-based timelines
  - No all risks are equal.
  - Timeline design must consider
    - Inherent risk level (likelihood × impact)
    - Exposure duration – how long the organization has been exposed
    - Risk velocity – how fast incidents could materialize
    - Compensating controls – can we “buy time” safely?
  - Critical, high-velocity risks require days, not months

TIMELINES

- Regulatory alignment
  - Certain timelines are effectively non-negotiable due to law or regulation
    - Patching deadlines for critical vulnerabilities in regulated industries
    - Time-bound remediation of audit findings
    - Contractual remediation SLAs with clients
  - Mitigation plans must explicitly reference these obligations and prioritize accordingly
  - Example
    - “Remediate high CVSS vulnerabilities within 30 days, critical within 7 days”
    - “Address audit-identified high-severity control deficiencies within 90 days”

TIMELINES

- Explicit milestones
  - Rather than a single end-date, mature plans define intermediate milestones, such as
    - Design complete by Week 2
    - Test environment implemented by Week 4
    - Pilot deployment completed by Week 6
    - Full production rollout by Week 8
    - Verification and closure by Week 10
  - Milestones allow
    - Progress tracking
    - Early detection of delays
    - Management visibility
  - This is where risk mitigation intersects with project management discipline

SUCCESS CRITERIA

- Define what “done” means
  - Whether risk has actually been reduced, not just “activities performed”
  - Good success criteria are
    - Specific: tied to a defined metric or state
    - Measurable: quantifiable with data
    - Auditable: evidence can be produced
    - Risk-linked: clearly reduce likelihood/impact

SUCCESS CRITERIA

- Example 1
  - “Reduce critical vulnerability count by 80% within 60 days”
    - *Metric*: # of critical vulnerabilities from scanner X
    - *Baseline*: 500 critical vulns today
    - *Target*: 100 or fewer in 60 days
    - *Risk Link*: Direct reduction of attack surface
    - *Auditability*: Before/after scan reports

SUCCESS CRITERIA

- Example 2
  - “Achieve 100% MFA enforcement on privileged accounts.”
    - *Metric*: % of privileged identities requiring MFA.
    - *Scope*: Admin accounts, root users, domain admins, cloud privileged roles.
    - *Target*: 100%. No exceptions unless formally approved and documented.
    - *Risk Link*: Reduces credential-based compromise risk.
    - *Auditability*: Identity provider logs, configuration evidence.

SUCCESS CRITERIA

- Example 3
  - “Reduce MTTR from 4 hours → 1 hour”
    - *Metric*: Mean Time to Recover (MTTR) for SEV-1 incidents
    - *Baseline*: 4 hours average over last quarter
    - *Target*: 1 hour in next quarter
  - Mitigation components
    - Automation (SOAR, scripts)
    - Better on-call coverage
    - Improved diagnostics and runbooks
    - *Risk Link*: Limits impact duration and financial loss.
    - *Auditability*: Incident logs & metric dashboards.

SUCCESS CRITERIA

- Example 4
  - “Zero backup integrity failures for 90 days”
    - *Metric*: # of failed restore tests or corrupt backups
    - *Baseline*: e.g., 5 integrity failures in past 90 days
    - *Target*: 0 failures in the next 90 days
  - Mitigation components
    - Better backup procedures
    - Verification scripts
    - Storage configuration fixes
    - *Risk Link:* Reduces data loss and recovery risk
    - *Auditability*: Backup reports and restore test logs

SUCCESS CRITERIA

- Why success criteria matter
  - Without solid success criteria
    - Mitigation degenerates into “activity theater”
    - Risk doesn’t actually change even though teams are busy
    - Leadership can’t tell if investments are reducing exposure
  - With clear criteria, risk and leadership can evaluate
    - Was the risk meaningfully reduced?
    - Did we hit deadlines?
    - Do we accept residual risk or plan further mitigation?

RESPONSIBLE PARTIES (RACI)

- If everyone is responsible, no one is responsible
  - Clear ownership is critical under pressure and for sustained follow-through
  - The RACI model in mitigation
    - Responsible (R) – Does the work
    - Accountable (A) – Ultimately answerable; approves completion
    - Consulted (C) – Provides expertise, must be engaged
    - Informed (I) – Kept updated; doesn't influence day-to-day actions

RESPONSIBLE PARTIES (RACI)

- Responsible
  - Ops engineer implementing failover
  - Security engineer deploying new EDR rules
  - Cloud engineer reconfiguring IAM or network
- Accountable
  - Product or system owner (e.g., Head of Payments Platform)
  - Ultimately “owns” the risk and signs off that mitigation is complete
- Consulted
  - Security team (control design and validation)
  - Risk and GRC (alignment with risk appetite and frameworks)
  - Legal and compliance (for regulatory considerations)
- Informed
  - Senior leadership (for major risk items)
  - Internal audit (for future assurance work)
  - Board-level committees (for strategic risks)

RESPONSIBLE PARTIES (RACI)

- Example: RACI for a Critical Vulnerability Mitigation
  - Risk: Critical remote code execution vulnerability in externally exposed web app
  - R – Responsible
    - App dev team (apply fix)
    - DevOps team (deploy patch)
  - A – Accountable
    - Application owner (business owner of that service)
  - C – Consulted
    - Security engineering (validate patch effectiveness)
    - Architecture team (ensure patch aligns with standards)
  - I – Informed
    - CISO, CIO
    - Risk committee

HIGH-QUALITY MITIGATION PLAN

- Example
- Risk: Elevated credential theft risk due to lack of MFA on privileged accounts.
  - Strategy: Mitigate via enforcing MFA and reducing unnecessary privileged accounts.
  - Resources:
    - Technical: IAM engineer, security engineer, directory services specialist
    - Financial: Budget for MFA token licensing
    - Infrastructure: Directory sync testing environment
  - Timeline:
    - Design & pilot: 2 weeks
    - Rollout to admins: 4 weeks
    - Rollout to service accounts where possible: 6 weeks
  - Success Criteria:
    - 100% of human privileged accounts protected by MFA within 6 weeks
    - 0 unprotected privileged logins detected in SIEM over 30-day verification period
  - RACI:
    - R: IAM engineer & security engineer
    - A: Head of Infrastructure
    - C: CISO, Security Architect
    - I: CIO, Internal Audit

HIGH-QUALITY MITIGATION PLAN

- Example
  - Risk: Elevated credential theft risk due to lack of MFA on privileged accounts
  - Strategy: Mitigate via enforcing MFA and reducing unnecessary privileged accounts
  - Resources
    - Technical: IAM engineer, security engineer, directory services specialist
    - Financial: Budget for MFA token licensing
    - Infrastructure: Directory sync testing environment
  - Timeline
    - Design & pilot: 2 weeks
    - Rollout to admins: 4 weeks
    - Rollout to service accounts where possible: 6 weeks

HIGH-QUALITY MITIGATION PLAN

- Success Criteria
  - 100% of human privileged accounts protected by MFA within 6 weeks
  - 0 unprotected privileged logins detected in SIEM over 30-day verification period
- RACI
  - R: IAM engineer & security engineer
  - A: Head of Infrastructure
  - C: CISO, Security Architect
  - I: CIO, Internal Audit

FALLBACK / CONTINGENCY PLANS

- Mitigation plans are designed to reduce risk by addressing root causes
  - But in real-world systems, especially complex, distributed, and tightly coupled ones, mitigation may
    - Take longer than expected
    - Only partially succeed
    - Be blocked by external constraints (vendors, regulations, dependencies)
    - Introduce new or unforeseen side effects
  - Fallback and contingency plans provide the safety net when mitigation is insufficient or too slow.
  - Core components of resilience engineering
    - They ensuring that critical services continue to operate, perhaps in degraded form, even when normal controls or mitigation paths are compromised

FALLBACK / CONTINGENCY PLANS

- Operational workarounds and alternate pathways
  - Fallback plans are tactical
  - Short to medium term responses
  - Allow the organization to maintain some level of service despite failures in primary systems or processes
  - Assumes that
    - The primary path is temporarily unavailable or unreliable
    - Full root-cause resolution will take time
    - The organization must continue operating in the meantime
  - Fallbacks are deliberately designed into architectures and processes

FALLBACK / CONTINGENCY PLANS

- Failover to backup systems
  - Failover mechanisms reroute workloads from a failing primary system to a secondary one
  - Examples
    - Database failover from primary region to a hot standby in another region
    - Switching from primary message queue cluster to a backup cluster
    - Moving traffic from one data center to a mirrored facility
  - Design considerations
    - Data consistency: Is the backup RPO acceptable (e.g., 5 minutes of data loss)?
    - Performance: Can the backup handle full production load?
    - Automation: Is failover automatic (health checks, DNS failover) or manual (runbook-driven)?
  - Failover is often a planned fallback, tested through BC/DR exercises

FALLBACK / CONTINGENCY PLANS

- Switch from automated to manual processing
  - When automated systems fail or become unreliable
  - Manual processes can serve as temporary bridges
  - Examples
    - Manual approval of high-value transactions when rule engines or fraud systems are offline
    - Customer service agents manually entering orders when front-end portals fail
    - Operations teams manually running batch jobs that normally run via scheduler
  - Risks and constraints
    - Human capacity limits (cannot scale indefinitely)
    - Higher error rates
    - Extended processing times
  - Fallback planning must define how long manual mode is sustainable and what throughput is realistic

FALLBACK / CONTINGENCY PLANS

- Reroute traffic to alternate regions
  - In cloud and distributed systems, fallback often involves redirecting traffic
  - Examples
    - Routing users in Region A to Region B during an outage
    - Using global load balancers to steer traffic away from failing zones
    - Redirecting authentication to a secondary identity cluster
  - Challenges
    - Data residency/privacy constraints (e.g., cross-border data flow)
    - Latency increases affecting user experience
    - Capacity planning in alternate regions
  - Fallback here is often tightly coupled with multi-region architecture and BC/DR strategies

CONTINGENCY PLAN

- Fallback plans are more operational and tactical
- Contingency plans are strategic, high-level plans
  - For dealing with severe, persistent, or systemic failures when
    - Primary and backup options are compromised
    - External dependencies fail dramatically (e.g., vendor outage, cloud provider failure)
    - Safety, legal, or regulatory concerns override normal operations
- Contingency plans often involve major shifts in how or where operations are conducted

CONTINGENCY PLAN

- Activate alternative suppliers or vendors
  - If a critical third-party becomes unavailable or unreliable
  - Contingency may involve shifting to alternatives
  - Examples
    - Switching payment processor due to an extended outage
    - Moving from a compromised cloud security provider to another vendor
    - Activating a backup logistics company in a supply chain disruptio.
  - Prerequisites
    - Pre-negotiated contracts or frameworks
    - Tested integration path
    - Risk assessments of alternative providers
  - This ties directly into third-party risk management and vendor resilience

CONTINGENCY PLAN

- Transition operations to a disaster recovery (DR) site
  - When primary environments are severely compromised, contingency may involve full DR activation
  - Examples
    - Moving core banking systems to a DR data center after a catastrophic event
    - Running all workloads from an alternate region for an extended period
    - Operating in DR mode after a ransomware event while restoring primary systems
  - Key issues
    - DR environments must be kept in sync (data, config, security posture)
    - Staff must be trained to operate from DR sites
    - Operating for extended periods in DR may incur substantial costs
  - This is where BC/DR and contingency strategy fully overlap

CONTINGENCY PLAN

- Switch authentication to offline mode
  - In identity crises (e.g., IdP outage, SSO failures, compromised SSO integration) may require some sort of contingency plan for authentication
  - Examples
    - Temporarily allow cached or local authentication with strict constraints
    - Use one-time codes or alternate identity verification channels (e.g., helpdesk-initiated)
    - Limit access to only those who can authenticate via a secondary method
  - Risk trade-off
    - Security risk increases (less robust authentication)
    - Operational risk decreases (users can log in)
    - Contingency decisions here must be risk-informed and time-bound

CONTINGENCY PLAN

- Execute emergency “Stop” procedures for unsafe conditions
  - Sometimes, the safest action is to halt operations
  - Examples
    - Stopping trading due to severe pricing or risk system malfunction
    - Pausing critical industrial processes in energy or manufacturing when safety systems are degraded
    - Halting high-value or high-risk transaction types (e.g., international wires) until controls are restored
  - This is akin to a “kill switch” and must be
    - Pre-authorized at the right level of leadership
    - Clearly documented and rehearsed
    - Supported by clear criteria and communication plans
  - In many regulated or safety-critical environments, failing to stop unsafe operations is more serious than temporary downtime

TRIGGERS

- Mark when fallbacks and contingencies activate
  - Fallbacks and contingencies must not depend on arbitrary judgment alone
  - Triggering conditions should be predefined, risk-based, and operationally clear
  - Many incidents start with attempts at standard mitigation
  - Fallbacks/contingencies kick in when
    - Mitigation is taking longer than RTO permits
    - There is no clear ETA for resolution
    - Complex root-cause analysis is ongoing and cannot be rushed
  - Example
    - Mitigation for an authentication bug is estimated at 6 hours; RTO is 1 hour
    - Trigger fallback (reduced functionality or alternate auth), while mitigation continues in parallel

TRIGGERS

- KRIs indicate escalating instability
  - KRIs can act as early-warning triggers for fallback/contingency
    - Rising error rates in a core service beyond dynamic thresholds
    - Increasing “near misses” in transaction processing
    - Accelerating capacity saturation and queuing delays
  - Example
    - KRI: Payment failure rate > 5% and rising
    - Trigger fallback: temporarily route payment traffic to alternate payment provider or degrade non-critical functions

TRIGGERS

- RTO/RPO thresholds are at risk
  - When the time since incident onset approaches RTO
  - Potential data loss is approaching RPO
  - Fallback and contingency must activate to avoid breach of resilience commitments
  - Example
    - RTO = 2 hours for core banking services
    - After 90 minutes of failed mitigation, risk team triggers contingency DR site activation or reduced functionality mode
  - Here, timeliness is non-negotiable

TRIGGERS

- Customer-facing SLAs are compromised
  - SLAs (latency, uptime, transaction success) often define
    - Contractual obligations
    - Financial penalties
    - Reputational risk thresholds
  - Triggered when live metrics show SLA breaches or imminent breaches
  - Example
    - Latency for core API > 2s for more than 10 minutes
    - Uptime < 99.9% for the current period
  - Fallback might
    - Disable secondary features
    - Increase rate-limiting on non-critical clients
    - Trigger traffic rerouting to reduce load

SAFETY NETS

- High-resilience organizations treat fallback and contingency planning as dynamic
  - Plans are updated after major incidents and Post-Incident Reviews
  - New attack vectors or failure modes result in new fallback/contingency patterns
  - Automation increasingly orchestrates fallback activation (e.g., automatic failover, circuit breakers, traffic shifting)
  - Exercises include forced activation of fallback/contingency to validate feasibility and human readiness

BUSINESS CONTINUITY AND DISASTER RECOVERY

- Business continuity and disaster recovery (BC/DR)
  - Ensures critical services and business functions can withstand, adapt to, and recover from disruptive events
    - Including technical failures, cyberattacks, third-party outages, physical incidents, or large-scale crise
  - Business Continuity (BC): Keeps business processes running, even if in degraded mode.
  - Disaster Recovery (DR): Restores technology systems to a trusted, operational state.
    - BC focuses on what the business must be able to do
    - DR focuses on what the technology must be able to provide to support BR

ADAPTIVE SYSTEMS

- Self-monitoring: continuous telemetry and anomaly detection
  - Systems continuously emit and analyze telemetry such as
    - Latency, error rates, queue depth
    - Authentication patterns
    - Resource usage, saturation metrics
    - Configuration and policy drift
  - Self-monitoring means
    - Metrics and logs are first-class citizens
    - Anomaly detection (statistical and ML-based) is built-in
    - Signals are correlated across services and layers

ADAPTIVE SYSTEMS

- Self-healing: automated remediation routines
  - Self-healing means automated correction of certain classes of faults
    - Restarting crashed processes or containers.
    - Redeploying from a golden image when configuration drift is detected.
    - Replacing unhealthy nodes automatically.
    - Revoking tokens or disabling accounts detected as compromised.
  - Self-healing reduces MTTR without waiting for human intervention

ADAPTIVE SYSTEMS

- Self-adjusting: auto-scaling, rebalancing, failover
  - Self-adjusting systems
    - Scale horizontally under load (add more instances/pods)
    - Rebalance traffic to healthy nodes or regions
    - Shed non-critical load when under stress (graceful degradation)
    - Adjust concurrency limits and rate limits dynamically
  - Examples
    - Auto-scaling groups in cloud environments that increase capacity in response to CPU or response-time signals
    - Service meshes that reroute traffic around failing instances
    - These behaviors reduce the likelihood that a surge or partial failure escalates into a full outage

ADAPTIVE SYSTEMS

- Machine-assisted reasoning
  - AI/ML-assisted threat classification
  - Adaptive systems leverage AI/ML to
    - Classify alerts based on historical data and context
    - Score risk levels for events or entities (e.g., user risk scores)
    - Identify subtle patterns of attack (low-and-slow or multi-stage threats)
    - Suggest likely root causes and recommended actions
  - These capabilities augment human teams, particularly during high-volume or complex events

ORGANIZATIONAL ADAPTATION

- Cross-functional collaboration
  - Adaptive organizations break silos between
    - Security
    - Operations / SRE
    - Development
    - Risk and GRC
    - Legal and Compliance
    - Business owners
  - Response involves multi-disciplinary teams working from a shared situational picture, not separated by rigid boundaries

ORGANIZATIONAL ADAPTATION

- Rapid, evidence-based decision-making
  - Decisions are
    - Data-informed (based on KPIs, KRIs, telemetry, threat intel)
    - Time-bounded (decisions made in minutes, not days)
    - Guided by pre-agreed frameworks (risk appetite, playbook criteria)
  - Adaptive organizations invest in
    - Clear decision rights
    - Predefined thresholds
    - Real-time dashboards

ORGANIZATIONAL ADAPTATION

- Flattened communication channels
  - During high-severity events
    - Hierarchical communication slows response
    - Adaptive organizations empower direct, lateral communication across teams
  - This may involve
    - Dedicated incident channels
    - Real-time incident “war rooms”
    - Clear but lightweight approval chains

ORGANIZATIONAL ADAPTATION

- Dynamic allocation of staff and expertise
  - Adaptive response requires
    - Quickly pulling in experts from across the org
    - Reprioritizing non-critical work to free key people
    - Using on-call rotations and swarm models effectively
  - Organizations may maintain
    - Incident commander rosters
    - Domain expert lists (e.g., IAM, payments, data)
    - Surge capacity plans for major incidents

ORGANIZATIONAL ADAPTATION

- Continuous learning from incidents
  - Every incident (and near miss) becomes input into
    - Updated playbooks
    - Better KRIs and KPIs
    - Architecture improvements
    - Training and simulations
  - Move from repeating the same failures to building resilience through learning

EXAMPLES OF ADAPTIVE RESPONSE

- Auto-isolation of compromised endpoints
  - Endpoint detection and response (EDR) tool detects suspicious behavior
    - For example: ransomware-like encryption
  - Automatically isolates the endpoint from the network
  - Triggers SOAR workflows: ticket creation, alerting, forensic snapshot
- Risk impact
  - Limits lateral movement
  - Containment is immediate, reducing blast radius

EXAMPLES OF ADAPTIVE RESPONSE

- Automated throttling of suspicious network activity
  - Anomaly detection identifies spikes in traffic from certain IPs or geos, or unusual query patterns
  - System automatically throttles or blocks traffic according to predefined policies
  - Security and Ops teams are alerted to validate and tune behavior
- Risk impact
  - Reduces DDoS or data-exfiltration impact
  - Protects critical services while investigation proceeds

EXAMPLES OF ADAPTIVE RESPONSE

- Continuous reprioritization of risk remediation
  - Based on real-time KRIs
  - KRI dashboards highlight rising exposures (e.g., vulnerability clusters, failed controls, near misses)
  - Risk teams continuously adjust remediation queues and priorities
  - Automated workflows reassign tasks, adjust due dates, and notify owners
- Risk impact
  - Ensures mitigation efforts target the most critical and time-sensitive risks, not just what is “on the plan”

EXAMPLES OF ADAPTIVE RESPONSE

- Continuous Reprioritization of Risk Remediation
  - Based on real-time KRIs
  - KRI dashboards highlight rising exposures (e.g., vulnerability clusters, failed controls, near misses).
  - Risk teams continuously adjust remediation queues and priorities.
  - Automated workflows reassign tasks, adjust due dates, and notify owners.
- Risk impact:
  - Ensures mitigation efforts target the most critical and time-sensitive risks, not just what is “on the plan.”

EXAMPLES OF ADAPTIVE RESPONSE

- Continuous Reprioritization of Risk Remediation
  - Based on real-time KRIs
  - KRI dashboards highlight rising exposures (e.g., vulnerability clusters, failed controls, near misses).
  - Risk teams continuously adjust remediation queues and priorities.
  - Automated workflows reassign tasks, adjust due dates, and notify owners.
- Risk impact:
  - Ensures mitigation efforts target the most critical and time-sensitive risks, not just what is “on the plan.”

Q&A AND OPEN DISCUSSION