VMMM Examples and Use Cases

Version: v2.0.0

Release Date: 2026-01-29

Introduction

This companion guide is designed to support the practical application of the Vulnerability Management Maturity Model (VMMM). While the VMMM provides structured descriptions of maturity across defined capability domains, this document brings those levels to life through concrete examples and realistic use cases.

The guide serves three key purposes:

1. Clarify Maturity Levels with Practical Examples:

   For each domain in the VMMM, we provide specific examples at Levels 1 through 5. These examples demonstrate how organizations at varying stages of maturity might behave, operate, or respond within that domain. This makes it easier for teams to self-assess their current state, recognize patterns, and identify opportunities for growth.

2. Illustrate Real-World Scenarios with Use Cases:

   The use cases offer narrative examples drawn from common challenges in vulnerability management—such as missed SLAs, failure to detect shadow IT, or inconsistent risk acceptance. Each use case walks through how the same scenario might unfold differently depending on an organization's maturity. These stories provide context and relevance, helping teams relate the model to their day-to-day realities.

3. Support Interpretation and Stakeholder Alignment:

   This guide helps reduce ambiguity and improve shared understanding across technical teams, security leadership, and governance bodies. It supports onboarding, maturity assessments, gap analysis, and internal education efforts. Whether you're building your first VM program or refining a mature one, the examples and use cases can guide decision-making and strategic investment.

How to Use This Guide

- During Self-Assessment: Compare your organization's practices with the examples at each maturity level to validate placement and uncover inconsistencies across teams or regions.
- For Training and Communication: Use the examples and use cases to explain the model to stakeholders, operational teams, or executives in relatable terms.
- To Plan Improvements: Identify gaps between current and target states, then reference the higher-level examples for guidance on process, tooling, or integration improvements.

This document is not a checklist, but a reference tool to support structured, contextual, and risk-informed vulnerability management improvement across your organization.

Maturity Level Examples

The following section provides concrete, level-by-level examples for every domain in the Vulnerability Management Maturity Model (VMMM). These examples are designed to help interpret the model's descriptions by showing how organizations at different levels of maturity might behave in practice.

For each domain, five examples are provided—one for each maturity level (1 through 5). The examples reflect observable behaviors, common scenarios, or typical program characteristics found at that level. While not exhaustive or prescriptive, they illustrate what the VMMM's definitions can look like in real environments.

Use these examples to:

- Validate how your current practices align with each level
- Identify what's required to progress from one level to the next
- Calibrate scoring across teams or business units using a consistent reference point
- Enhance internal conversations by linking maturity language to operational realities

The examples were intentionally written to be neutral, non-tool-specific, and applicable across industries. They emphasize measurable behaviors, process maturity, and risk-aware decision-making. This section is especially useful for those facilitating maturity assessments or guiding improvement planning efforts.

Prepare Phase

Domain: Context

- *Level 1:* No consistent understanding of how business operations or regulatory obligations influence vulnerability management. Teams focus only on scanning and fixing without strategic direction.
- *Level 2:* Some teams consider external pressure (e.g., customer audits or compliance deadlines) but apply it inconsistently. Business alignment depends on individual initiative.
- *Level 3:* Vulnerability management policies reference key business functions and external risk drivers. Prioritization considers compliance obligations and departmental risk tolerances.
- *Level 4:* Business impact ratings are assigned to systems and factored into prioritization. Contextual data such as service criticality and customer impact is included in decision support tools.
- *Level 5:* Real-time business and threat context dynamically adjusts prioritization. Vulnerability dashboards align with business KPIs and are reviewed jointly by security and business leaders.

Domain: Crisis Response & Zero-Day Readiness

- *Level 1:* There is no defined process for handling zero-day vulnerabilities or high-impact campaigns. Teams scramble when urgent threats emerge.
- *Level 2:* Security leads manually coordinate response to major threats based on ad hoc triage. Past incidents reveal gaps in ownership and communication.
- *Level 3:* The organization has documented zero-day playbooks and named owners. Prior critical incidents have resulted in defined escalation steps and team responsibilities.
- *Level 4:* Cross-functional crisis simulations are conducted annually. Zero-day incidents are tracked in an incident register, and metrics are used to improve response speed.
- *Level 5:* Response readiness is tested quarterly. Playbooks are tailored by threat type, automatically triggered, and integrated with business continuity, communications, and executive reporting.

Domain: Policy & Standards

- *Level 1:* No formal VM policies or standards exist. Teams act independently with no defined requirements.
- *Level 2:* Basic patch SLAs or standards exist for some systems but are inconsistently applied. Documentation is minimal.
- *Level 3:* Organization-wide policies are defined, reviewed, and aligned with compliance requirements. Standards define ownership, timelines, and risk thresholds.
- *Level 4:* Policies and standards are embedded into workflows such as CI/CD pipelines and Infrastructure-as-Code templates, with automated enforcement. Exceptions are centrally documented.

- *Level 5:* Policies are continuously updated using metrics, threat intelligence, and business feedback. Enforcement is automated across all environments, and policy effectiveness is measured.

Domain: Program Governance

- *Level 1:* No formal governance structure for vulnerability management exists. Accountability and oversight are unclear.
- *Level 2:* Security leadership meets periodically to review vulnerability status but without defined governance responsibilities or escalation authority.
- *Level 3:* A governance body with defined roles and responsibilities oversees vulnerability management, meeting monthly to review metrics and issues.
- *Level 4:* Governance includes business, IT, and compliance participation. Remediation delays, exceptions, and systemic issues are reviewed and acted on with accountability.
- *Level 5:* Governance is integrated with enterprise risk structures, driving investment decisions and strategic alignment. Reports feed into board-level risk discussions.

Domain: Risk Appetite & Tolerance Definitions

- *Level 1:* No defined risk appetite or tolerance related to vulnerabilities. Teams make decisions ad hoc.
- *Level 2:* Some informal thresholds exist (e.g., patch critical vulns in X days), but they are inconsistently communicated.
- *Level 3:* Risk tolerance is documented and communicated, tied to severity and asset criticality. SLAs are aligned to these definitions.
- *Level 4:* Tolerance definitions include exploit-based thresholds (e.g., KEV listing or EPSS score triggers accelerated timelines). Executives and governance review them periodically.
- *Level 5:* Risk appetite is continuously refined with input from governance, business context, exploit data, and performance metrics. Thresholds are adaptive and integrated into risk dashboards.

Domain: Security Ecosystem Integration

- *Level 1:* Vulnerability data is siloed and not connected to other systems. Asset, threat, and remediation systems operate independently.
- *Level 2:* Vulnerability data is exported manually to other teams, but workflows are fragmented. Integration depends on individuals.
- *Level 3:* Vulnerability tools are integrated with asset inventory and ticketing systems. Remediation tickets are auto-generated based on scan data.
- *Level 4:* Bidirectional integration exists across threat detection, CMDB, change management, and SIEM platforms. Contextual data enriches risk decisions.
- *Level 5:* The VM program is embedded in the broader security fabric. Integration enables closed-loop workflows, real-time visibility, and automated decisioning.

Domain: VM Roles & Responsibilities

- *Level 1:* No assigned roles for vulnerability management exist. Responsibilities are informal and often ignored.
- *Level 2:* Certain individuals are known to "own" vulnerability tasks but without formal documentation. Coverage is inconsistent.
- *Level 3:* Defined roles and responsibilities exist for all stages of vulnerability management. Accountability is assigned by system or function.
- *Level 4:* Role clarity is embedded into operational processes, with ownership tracked through metrics and supported by management.
- *Level 5:* Roles are evaluated and adjusted as part of program reviews. Ownership is reinforced by KPIs and performance assessments.

Domain: Crisis Communication Readiness

- *Level 1:* There is no plan for communicating about major vulnerabilities internally or externally. Messaging is reactive and inconsistent.
- *Level 2:* Security teams send informal updates when crises occur. There is no centralized communication plan or leadership coordination.
- *Level 3:* Communication plans exist for zero-day and high-impact vulnerability events. Templates are used for executive and stakeholder updates.
- *Level 4:* Crisis communication playbooks are tested during tabletop exercises. Communications are tailored to internal, external, and regulatory audiences.
- *Level 5:* Messaging is pre-approved for critical scenarios. Communication triggers are linked to response processes and include executive briefings, PR guidance, and legal review.

Domain: Data Quality & Source of Truth

- *Level 1*: Different teams use scanner data, CMDB entries, and cloud accounts with no reconciliation. Reports conflict.
- *Level 2:* Manual spot-checks between scanner exports and CMDB happen quarterly. Many assets are still missed.
- *Level 3:* A central CMDB or asset system is the "source of truth." Reconciliation with scanners/cloud is tracked, though not fully automated.
- *Level 4:* Automated reconciliation enforces consistent asset IDs. Coverage metrics are reviewed in governance meetings.
- *Level 5:* Data quality is continuously monitored, self-healing processes correct inconsistencies, and asset data is enriched with business context.

Domain: Third-Party VM Readiness

- *Level 1:* Contracts have no VM requirements for vendors.
- *Level 2:* A few agreements mention "maintain secure systems," but no enforceable language.

- *Level 3:* Procurement templates include patch timelines and vulnerability notification clauses.
- *Level 4:* All contracts include standardized VM expectations (SBOMs, disclosure SLAs, patch SLAs). Evidence is reviewed periodically.
- *Level 5:* Supplier VM performance is continuously monitored. Data integrates into enterprise dashboards and influences vendor selection.

Identify Phase

Domain: Ephemeral & Short-Lived Asset Discovery

- *Level 1:* No mechanisms are in place to detect or assess containers, serverless functions, or other short-lived assets. These are often excluded from scans.
- *Level 2:* Teams occasionally scan containers or cloud assets manually, but short-lived systems often disappear before scans complete.
- *Level 3:* CI/CD pipelines trigger vulnerability scans during container builds. Short-lived assets are scanned upon deployment and captured in asset inventory.
- *Level 4:* Automated discovery tools continuously detect and scan ephemeral assets as they are spun up or terminated, using orchestration hooks.
- *Level 5:* Detection and assessment are integrated into IaC workflows, with telemetry feeding directly into risk dashboards and coverage metrics.

Domain: External Vulnerability Intelligence Ingestion

- *Level 1:* Teams rely solely on internal scan tools and do not monitor external vulnerability disclosures or threat intelligence feeds.
- *Level 2:* Critical CVEs are occasionally tracked using email alerts or vendor newsletters, but the process is manual and informal.
- *Level 3:* External sources like NVD, vendor advisories, and threat intel feeds are consumed regularly and inform vulnerability reviews.
- *Level 4:* External intel is enriched with asset context and exploitability to inform prioritization. Data is ingested into central platforms.
- *Level 5:* Threat intelligence is correlated in real time with internal vulnerabilities. Exploit maturity and exposure mapping are automated and influence ticket routing.

Domain: Manual Discovery & Analyst Testing

- *Level 1:* No manual testing is performed. Teams rely entirely on automated tools regardless of environment or complexity.
- *Level 2:* Ad hoc manual reviews happen when scan coverage is questioned, but there is no formal process or documentation.
- *Level 3:* Analysts are tasked with reviewing high-value systems periodically. Manual discovery is used to confirm scan accuracy and uncover logic flaws.
- *Level 4:* Manual discovery is integrated into standard testing workflows for crown-jewel systems. Results are tracked and used to tune tooling.
- *Level 5:* A structured manual testing program supplements automation. Findings are documented, validated, and used to improve detection capability and analyst training.

Domain: Shadow IT & Rogue Asset Detection

- *Level 1:* No attempts are made to detect unauthorized systems or cloud services. Shadow IT is discovered only after an incident.

- *Level 2:* Teams occasionally conduct manual IP sweeps or rely on reports from finance or procurement to spot unmanaged assets.
- *Level 3:* Network telemetry and procurement data are reviewed quarterly to identify rogue systems. Detected assets are investigated and cataloged.
- *Level 4:* EASM tools and internal logs are used continuously to identify unauthorized deployments. Alerts are reviewed by operations teams.
- *Level 5:* Shadow IT detection is fully integrated into onboarding, network monitoring, and incident processes. Unauthorized assets are automatically flagged and contained.

Domain: Third-Party Asset Discovery

- *Level 1:* There is no inventory of assets hosted or operated by third parties. These systems are invisible to vulnerability management.
- *Level 2:* Some vendor systems are listed during audit cycles, but there is no ongoing tracking or scanning.
- *Level 3:* A third-party asset register is maintained and reviewed semi-annually. Key vendor systems are included in vulnerability scans or assessments.
- *Level 4:* Contracts require asset discovery and vulnerability disclosures from vendors. Periodic attestations and spot checks are conducted.
- *Level 5:* External assets are discovered through attack surface management. Scanning is automated where permitted, and noncompliance is tracked through vendor governance.

Domain: Asset Inventory & Classification

- *Level 1:* The organization does not maintain a current asset inventory. Many systems are unmanaged or unknown to security teams.
- *Level 2:* Asset lists exist in silos across IT and security. Updates are manual and classification is inconsistent or absent.
- *Level 3:* A unified inventory exists, updated regularly, with asset classification based on ownership, type, and criticality.
- *Level 4:* Asset data is synchronized across discovery, CMDB, and vulnerability tools. Classification informs scan scheduling and remediation SLAs.
- *Level 5:* Inventory is enriched with business metadata and security context. Classifications drive prioritization, governance reporting, and risk scoring.

Domain: Automated Vulnerability & Exposure Scanning

- *Level 1:* Scans are ad hoc, often unauthenticated, with no coverage tracking.
- *Level 2:* Routine scans cover some systems monthly/quarterly. Cloud/containers often excluded.
- *Level 3:* Enterprise-wide scanning with authenticated coverage for critical platforms. Coverage metrics tracked.
- *Level 4:* Continuous scanning across hybrid environments. Integrated into CI/CD and container workflows.

- *Level 5:* Adaptive scanning informed by threat intel (KEV, EPSS). Includes EASM/SaaS coverage, feeding directly into risk dashboards.

Domain: Application & Service Discovery

- *Level 1:* No inventory of applications, APIs, or SaaS services.
- *Level 2:* Some critical apps are documented, but APIs and dependencies are missed.
- *Level 3:* Application inventory includes web apps, APIs, and SaaS. Basic SBOMs generated.
- *Level 4:* Automated discovery through API gateways and service mapping. SBOMs integrated into vulnerability analysis.
- *Level 5:* Continuous discovery across apps, APIs, SaaS, and dependencies. Automated SBOMs linked to prioritization and governance.

Analyze Phase

Domain: Business Impact Modeling

- *Level 1:* No effort is made to understand how vulnerabilities affect business outcomes. Risk is considered purely technical.
- *Level 2:* Teams occasionally consider business impact when systems are obviously critical, but this is not formalized.
- *Level 3:* Systems are categorized based on business function. Vulnerabilities on high-impact systems are prioritized based on downtime, revenue, or compliance risk.
- *Level 4:* Business impact ratings are integrated into vulnerability scoring and reporting. Risk assessments include stakeholder-defined criteria.
- *Level 5:* Impact models are refined continuously using operational data and stakeholder input. Risk scores directly influence investment decisions and mitigation campaigns.

Domain: Exploitability Assessment

- *Level 1:* Exploitability is not considered. All vulnerabilities are treated equally regardless of threat landscape.
- *Level 2:* Teams sometimes reference public exploit databases or vendor alerts, but there is no structured assessment.
- *Level 3:* Known exploit availability is factored into prioritization. Vulnerabilities are tagged when weaponized exploits exist.
- *Level 4:* Exploit maturity, prevalence, and attacker behavior are incorporated using EPSS or threat intel feeds. Internal exposure is used to tune urgency.
- *Level 5:* Exploitability scoring is fully automated and refreshed daily. Patterns in attacker exploitation are used to anticipate future risk and adjust defenses.

Domain: False Positive & Suppression Validation

- *Level 1:* Suppression rules are applied arbitrarily with no review. Teams suppress findings to reduce alert fatigue or meet patching metrics.
- *Level 2:* Suppressions are occasionally documented but rarely reviewed. No validation is done to confirm if findings were truly false.
- *Level 3:* Suppression criteria are documented. Security teams validate high-impact suppressions and maintain a review log.
- *Level 4:* Suppression review is a defined part of triage. Historical suppressions are audited quarterly and adjustments made based on findings.
- *Level 5:* Suppressions are tracked by category and correlated with risk data. Validation is automated for known false positives, with feedback loops into detection tools.

Domain: Risk-Based Prioritization

- *Level 1:* All vulnerabilities are treated equally or prioritized by CVSS score alone. No consideration is given to business impact or exploitability.

- *Level 2:* Teams manually adjust prioritization when major threats emerge, but decisions are inconsistent and undocumented.
- *Level 3:* Prioritization considers asset criticality, severity, and business function. Risk scoring models are applied across teams.
- *Level 4:* Prioritization integrates exploitability, asset value, exposure, and risk tolerance thresholds. Output feeds into ticketing queues.
- *Level 5:* Risk scoring is dynamic and continuously updated with threat intel, business context, and aging data. Prioritization logic is governed and monitored.

Domain: Third-Party Risk Identification

- *Level 1:* Third-party risk is not evaluated in vulnerability analysis. Vendors and partners are outside the scope of review.
- *Level 2:* Risk from vendors is acknowledged informally but not tracked. Action is only taken during major incidents.
- *Level 3:* Vendor-provided systems are reviewed for vulnerabilities. Risk is factored into exception reviews and remediation decisions.
- *Level 4:* Vulnerability impact is analyzed across internal and third-party boundaries. Issues with shared systems are escalated through vendor management.
- *Level 5:* Third-party risk is scored and included in enterprise risk dashboards. Correlations between vendor exposure and internal impact drive joint remediation efforts.

Domain: Root Cause Analysis

- *Level 1:* No root cause reviews are performed. Recurring vulnerabilities or failed remediations go unexamined.
- *Level 2:* Teams conduct informal reviews after repeated failures but do not document findings.
- *Level 3:* Root cause analysis is triggered after SLA violations or repeat issues. Findings are documented and corrective actions assigned.
- *Level 4:* RCA is integrated into post-mortems and included in governance reports. Themes are tracked over time to guide policy and control improvements.
- *Level 5:* RCA outcomes are measured and feed into program KPIs. Insights are used to proactively adjust detection, policy, and ownership models.

Domain: Threat Intelligence Correlation & Exploit Analysis

- *Level 1:* No correlation is performed. Exploitable vulnerabilities are treated the same as low-risk findings.
- *Level 2:* Teams manually flag issues with known exploits when time allows. Decisions rely on individual analyst judgment.
- *Level 3:* Threat intelligence is reviewed weekly to identify CVEs with known exploitation. Correlation guides remediation prioritization.
- *Level 4:* Exploitability scoring (e.g., EPSS or threat feeds) is integrated into dashboards and decision logic.

- *Level 5:* Exploit analysis is automated, tracked over time, and tied to remediation outcomes. High-risk items are surfaced for escalation and cross-functional review.

Domain: Vulnerability Aging & Exposure Tracking

- *Level 1:* No tracking of vulnerability age or exposure is performed. Teams are unaware how long vulnerabilities remain unresolved.
- *Level 2:* Aging is sometimes calculated manually for audits or reports, but not used to guide decisions.
- *Level 3:* Aging metrics are reported monthly and reviewed by governance. SLA breaches are flagged for follow-up.
- *Level 4:* Exposure duration is tracked in real time and tied to asset criticality. Metrics influence prioritization and resourcing.
- *Level 5:* Aging trends are benchmarked across teams and environments. High-exposure areas trigger proactive remediation campaigns and leadership reviews.

Domain: Vulnerability Clustering & Campaigns

- *Level 1:* Vulnerabilities are addressed one by one. No grouping or coordinated efforts are used.
- *Level 2:* Occasionally teams group issues by software version or vendor during major patching pushes.
- *Level 3:* Clusters are used to create remediation campaigns for recurring issues or shared technologies. Campaign progress is tracked.
- *Level 4:* Campaigns are driven by themes such as privilege escalation or supply chain. Dashboards report campaign progress and risk reduction.
- *Level 5:* Clustering is automated. Campaigns are prioritized based on business impact, risk trends, and stakeholder engagement.

Communicate Phase

Domain: Alerting & Operational Notification

- *Level 1:* Vulnerability findings are stored in scan tools with no alerting. Operational teams are unaware of issues unless manually informed.
- *Level 2:* Email alerts are sent manually for critical vulnerabilities, but alerting is inconsistent and often missed.
- *Level 3:* Alerts are sent automatically to asset owners or ticketing systems. Notifications are based on severity or asset type.
- *Level 4:* Alerting thresholds are defined and tailored by system criticality. Teams receive actionable alerts with context and resolution paths.
- *Level 5:* Alerts are risk-scored and prioritized dynamically. Operational dashboards show open alerts, time to acknowledgment, and SLA status.

Domain: Exception & Risk Acceptance Communication

- *Level 1:* Exceptions are granted verbally or through informal agreements. There is no visibility into what risks have been accepted.
- *Level 2:* Some exceptions are recorded in spreadsheets but rarely shared beyond the team that requested them.
- *Level 3:* Approved exceptions are logged in a central repository and reported monthly. Business owners are copied on acceptance notices.
- *Level 4:* Risk acceptance communications are structured, tracked, and included in quarterly governance reports.
- *Level 5:* Exceptions are communicated to affected stakeholders, with full context and expiry dates. Business unit leaders review risk exposure routinely.

Domain: Governance & Escalation Reporting

- *Level 1:* There is no formal escalation process for unresolved vulnerabilities or systemic failures.
- *Level 2:* Escalations happen reactively when major problems arise, but are undocumented and inconsistent.
- *Level 3:* Escalation criteria and pathways are defined. Reports are generated for overdue issues or repeated failures.
- *Level 4:* Escalated items are tracked to resolution. Governance committees review trends and systemic breakdowns.
- *Level 5:* Escalation patterns inform program investments. Persistent issues trigger root cause reviews and are logged in risk registers.

Domain: Metrics & Performance Reporting

- *Level 1:* No consistent metrics are reported. Status is conveyed informally or not at all.

- *Level 2:* Basic charts or spreadsheets are created manually when requested. There is no standard reporting cadence.
- *Level 3:* Standard metrics (e.g., SLA compliance, scan coverage, vulnerability aging) are reported monthly to stakeholders.
- *Level 4:* Dashboards provide role-based views and trend analysis. Reports inform governance discussions and risk prioritization.
- *Level 5:* Metrics are aligned to business goals and KPIs. Reporting includes forecasting, benchmarking, and performance improvement plans.

Domain: Stakeholder Engagement & Risk Framing

- *Level 1:* Risk is communicated in technical terms with no translation for non-technical audiences.
- *Level 2:* Security teams sometimes prepare executive summaries, but content lacks business alignment or clarity.
- *Level 3:* Risk is framed using business language and linked to potential operational or compliance impact. Reports are tailored to stakeholder roles.
- *Level 4:* Engagement is structured through regular briefings, with shared accountability for decision-making and trade-offs.
- *Level 5:* Risk communication supports strategic planning. Framing is dynamic, data-driven, and informs cross-functional investment decisions.

Domain: Vulnerability Disclosure Management

- *Level 1:* The organization has no defined process for external vulnerability disclosure. Researcher reports or bug submissions are ignored or handled ad hoc. Customers and partners are not notified when vulnerabilities are identified.
- *Level 2:* A basic intake mechanism exists (e.g., a security@ mailbox or web form), but reports are inconsistently acknowledged or tracked. Advisories are rare, reactive, and lack standardization.
- *Level 3:* A published Vulnerability Disclosure Policy (VDP) exists. Intake and triage processes follow defined SLAs, and advisories are published using standardized templates. Roles for security, legal, and PR are defined.
- *Level 4:* A Product Security Incident Response Team (PSIRT) coordinates disclosures across business units. Advisories and CVE assignments follow repeatable workflows aligned with patch or mitigation releases. Coordinated disclosure with researchers is standard practice.
- *Level 5:* Vulnerability disclosure management is proactive, transparent, and metrics-driven. Automation supports intake, case tracking, and researcher communication. Metrics such as time-to-acknowledge, disclosure timelines, and advisory effectiveness are reviewed. Lessons learned are fed into governance and prevention processes.

Treatment Phase

Domain: Change Management Integration

- *Level 1:* Remediation tasks are carried out without coordination with change management. Risk is introduced through untracked changes.
- *Level 2:* Major patches are sometimes discussed in change meetings, but vulnerability response is not linked to formal processes.
- *Level 3:* Critical vulnerability fixes are logged in the change management system. Emergency changes follow a defined approval path.
- *Level 4:* Change management policies include guidance for vulnerability-driven changes. Risk reviews are triggered for delayed fixes.
- *Level 5:* Vulnerability remediation is embedded into change workflows. Dashboards show pending, approved, and completed remediation activities tied to change tickets.

Domain: Compensating Controls

- *Level 1:* No compensating controls are documented or applied. Vulnerabilities remain unresolved without mitigation.
- *Level 2:* Temporary measures like firewall blocks or user restrictions are applied, but not formally approved or tracked.
- *Level 3:* Controls are proposed and reviewed during exception handling. Documentation includes rationale and scope.
- *Level 4:* Compensating controls are cataloged, monitored, and linked to risk acceptance or deferred remediation.
- *Level 5:* Effectiveness of controls is measured. Reviews occur quarterly, and controls influence risk scoring and program metrics.

Domain: Configuration Management

- *Level 1:* There is no enforcement of secure configurations. System hardening is left to individual teams.
- *Level 2:* Some baseline configurations are published but not consistently applied or audited.
- *Level 3:* Configuration baselines are enforced for critical systems. Vulnerability scans include misconfiguration findings.
- *Level 4:* Configurations are monitored using automated compliance tools. Deviations are tracked and addressed within SLAs.
- *Level 5:* Secure configurations are tailored by system type and integrated into CI/CD, golden images, and IaC pipelines.

Domain: Patch Management

- *Level 1:* Patching is irregular and left to each team. There is no visibility into coverage or timeliness.
- *Level 2:* Patching occurs on a basic schedule but lacks coordination with security findings or asset criticality.
- *Level 3:* Patch SLAs are defined by severity and enforced for key platforms. Coverage is reported monthly.
- *Level 4:* Patch cycles are prioritized by vulnerability risk and asset value. Metrics track compliance and delays.
- *Level 5:* Patching is adaptive and driven by real-time prioritization. Patch readiness is assessed during asset provisioning.

Domain: Remediation Orchestration & Automation

- *Level 1:* All remediation is manual. Security teams email issues to operations with no follow-up tracking.
- *Level 2:* Teams create tickets for high-severity issues, but there is no orchestration or consistent handoff.
- *Level 3:* Remediation is orchestrated through workflows that assign and track tickets. High-risk vulnerabilities are routed based on asset owner and criticality.
- *Level 4:* Playbooks automate ticket generation, routing, and status updates. APIs link security platforms and ITSM tools.
- *Level 5:* Remediation is adaptive and governed by policies. Closed-loop automation adjusts actions based on risk, ownership, and real-time status.

Domain: Remediation Validation & Closure

- *Level 1:* Fixes are assumed complete after patching. No validation occurs.
- *Level 2:* Teams occasionally rescan systems after remediation, but it is not standard practice.
- *Level 3:* Validation scans confirm issue resolution before closure. Failed fixes trigger rework tickets.
- *Level 4:* Closure requires validation through scan, logs, or system attestations. All remediation is tracked to closure.
- *Level 5:* Validation is continuous and automated. Dashboards show closure rates, false fixes, and effectiveness over time.

Domain: Risk Acceptance Governance

- *Level 1:* Anyone can choose not to fix a vulnerability. Decisions are undocumented.
- *Level 2:* Some exceptions are submitted to security, but approvals are inconsistent.
- *Level 3:* A defined process exists for accepting risk, with documented rationale, timelines, and business owner signoff.

- *Level 4:* Exceptions are reviewed periodically, tracked in dashboards, and tied to asset criticality and risk thresholds.
- *Level 5:* Risk acceptance is monitored by governance. Patterns of acceptance influence strategic remediation priorities and program investments.

Use Cases

This section presents real-world use cases aligned to each domain of the Vulnerability Management Maturity Model (VMMM). Each use case illustrates how the same scenario might unfold across different levels of maturity—from unstructured and reactive practices to fully integrated and optimized approaches.

These use cases are designed to:

- Demonstrate the impact of maturity by showing the consequences of low vs. high maturity in real operational situations
- Make abstract model concepts more relatable by connecting them to recognizable problems, such as delayed patching, risk acceptance gaps, or failure to detect rogue assets
- Support stakeholder alignment and communication by providing narrative examples that can be shared across technical, governance, and business teams
- Guide program improvement by helping teams reflect on how similar scenarios might be handled today—and what higher maturity would enable

Each use case includes:

- A short scenario drawn from realistic vulnerability management challenges
- Descriptions of how organizations at each maturity level might handle the situation
- A summary of the outcomes and benefits realized at higher levels of maturity

These narratives are meant to complement, not replace, the level-by-level descriptions and examples found earlier in the document. While every organization's context is unique, these use cases can serve as conversation starters, training materials, or a reference point during strategic planning and maturity assessment discussions.

Prepare

Use Case: Responding to a High-Impact Zero-Day Without Established Crisis Procedures

Domain: Crisis Response & Zero-Day Readiness

Scenario:

A zero-day vulnerability affecting a core infrastructure component (such as a remote code execution flaw in a widely deployed edge device) is disclosed by a vendor with no immediate patch available. Exploits are reported in the wild within hours of the announcement. The organization has hundreds of exposed devices, including systems supporting customer-facing operations.

Action at Different Maturity Levels:

- **Level 1:** The security team scrambles to react. There is no established escalation or playbook. Response actions vary by team and location. Leadership is notified late.
- **Level 2:** A small team attempts to coordinate through email and ad hoc meetings. Response actions (e.g., temporary firewall rules) are applied inconsistently. There is no structured follow-up.
- **Level 3:** A crisis response playbook is triggered. Roles and responsibilities are pre-defined. Emergency meetings are held with IT and security. Leadership is briefed, and mitigation guidance is disseminated internally.
- **Level 4:** Threat intel, asset inventories, and detection tooling are consulted to scope impact within hours. Cross-functional response teams coordinate using a formal crisis workflow. Communication templates are used for internal updates.
- **Level 5:** Crisis response is practiced through periodic exercises. Impacted systems are immediately flagged through tagging and telemetry. Response actions, communication, and compensating controls are executed within service-level timelines. Lessons learned are logged and integrated into future readiness improvements.

Outcome:

Organizations at higher maturity levels reduce uncertainty, coordinate faster, and communicate more effectively—minimizing both technical exposure and reputational risk. Executive stakeholders receive timely updates, and resource allocation is directed where the risk is highest.

Use Case: Conflicting Vulnerability Management Policies Across Business Units Create Compliance Risk

Domain: Policy & Standards

Scenario:

A multinational technology company operates with independently managed regional IT teams. Each region has developed its own patch management timelines and exception approval processes. During a compliance audit, regulators discover that critical systems in Europe are patched within 30 days while identical systems in North America remain unpatched for 90+ days. The lack of unified policy creates inconsistent risk posture and audit findings across the enterprise.

Action at Different Maturity Levels:

- **Level 1:** No formal policies exist. Each team patches based on convenience or individual judgment. Audit findings reveal widespread inconsistency with no documented standards.
- **Level 2:** Basic patching guidelines exist in some regions but are not enforced. Documentation is scattered across wikis and emails. Exception processes vary by manager.
- **Level 3:** Enterprise-wide VM policies are documented and published. Patch timelines are defined by severity, and exception processes require written justification. Policies are reviewed annually.
- **Level 4:** Policies are embedded into automated workflows. CI/CD pipelines enforce configuration standards. Policy violations trigger alerts, and exceptions are tracked centrally with expiration dates.
- **Level 5:** Policies are continuously refined using program metrics and threat intelligence. Automated enforcement spans all environments. Policy effectiveness is measured through compliance rates, exception volumes, and time-to-remediation trends reported to governance.

Outcome:

Organizations with mature policy frameworks ensure consistent risk management across all business units and geographies. Clear, enforceable standards reduce audit friction, improve accountability, and enable scalable vulnerability management. At higher maturity, policies evolve with the threat landscape and operational realities, balancing security requirements with business agility.

Identify

Use Case: Discovering Short-Lived Assets in a Containerized Environment

Domain: Ephemeral & Short-Lived Asset Discovery

Scenario:

A development team deploys microservices using containers that spin up and shut down rapidly. A critical vulnerability in the base container image goes unnoticed in production because the asset scanner runs on a weekly schedule and cannot capture short-lived workloads. The issue is discovered only after a third-party penetration test.

Action at Different Maturity Levels:

- **Level 1:** Short-lived assets are not part of the asset inventory. Teams have no visibility and assume that weekly scans are sufficient.
- **Level 2:** Scans are sometimes triggered manually during deployment, but teams struggle to capture containers in time.
- **Level 3:** CI/CD pipelines integrate vulnerability scans into the build stage. Containers are checked before deployment but not monitored at runtime.
- **Level 4:** Runtime orchestration tools detect container startups and trigger lightweight scanning. Asset inventory is updated dynamically.
- **Level 5:** The container registry, CI/CD, and orchestration platforms are fully integrated. Vulnerabilities are detected pre-deployment and correlated in real time during runtime using ephemeral asset tags.

Outcome:

Higher maturity enables full visibility into transient workloads, ensuring vulnerabilities are identified and addressed before reaching production. Missed windows of exposure are eliminated, and risk from volatile assets is minimized.

Use Case: Identifying Shadow IT in a Cloud-Centric Workforce

Domain: Shadow IT & Rogue Asset Detection

Scenario:

A regional sales team deploys a third-party CRM tool hosted in a public cloud to manage client contacts, bypassing corporate IT. The system stores sensitive data but is not covered by vulnerability scans, MFA policies, or the corporate firewall. A data breach from this shadow system exposes confidential client information.

Action at Different Maturity Levels:

- **Level 1:** The organization is unaware the asset exists. Discovery occurs only after an incident.
- **Level 2:** A one-time internal audit surfaces the system, but it is not formally addressed or tracked afterward.
- **Level 3:** Network logs and DNS records are reviewed monthly to detect unsanctioned systems. The rogue CRM is flagged for review.
- **Level 4:** External asset management and cloud monitoring tools identify new SaaS usage in real time. Alerts are sent to security teams.
- **Level 5:** Shadow IT is automatically detected and categorized using telemetry, identity systems, and purchasing data. Asset owners are notified, and risk is logged in the central governance system.

Outcome:

With increasing maturity, the organization gains the ability to rapidly discover and respond to rogue deployments, reducing regulatory exposure and protecting sensitive data across uncontrolled environments.

Analyze

Use Case: Applying Risk-Based Prioritization to a Critical Vulnerability in a Business-Critical Application

Scenario:

The vulnerability management team receives a high-severity CVE (CVSS 9.8) affecting a widely used Java library. Initial scanning reveals its presence in multiple systems across the organization, including a public-facing web application that handles client transactions and several internal development tools.

Action at Different Maturity Levels:

- **Level 1:** All systems with the CVE are treated equally. The team starts with low-effort systems, delaying remediation for the business-critical web application.
- **Level 2:** The team is aware of the external exposure and manually prioritizes the web application. The rationale is not documented, and internal tools remain unaddressed.
- **Level 3:** A risk model is applied. The public-facing application is flagged as high priority due to business value and external exposure. The vulnerability is tracked through ticketing and assigned an SLA.
- **Level 4:** Risk-based prioritization includes exploitability (EPSS score > 0.85), asset classification, and exposure window. The web app is patched within 24 hours, while internal dev tools are scheduled within the risk threshold.
- **Level 5:** Prioritization is automated. Business impact, exploit data, and patch readiness dynamically route the vulnerability to the correct teams. The web application fix is deployed and validated in under 12 hours, with risk dashboards updated in real time.

Outcome:

At higher maturity, resources are allocated efficiently based on business risk, and the most exposed systems are remediated fastest. Risk is documented, communicated, and reduced with minimal disruption.

Use Case: Failing to Prioritize an Actively Exploited Vulnerability

Domain: Threat Intelligence Correlation & Exploit Analysis

Scenario:

A critical vulnerability affecting a widely used VPN appliance is disclosed publicly. Within 48 hours, CISA adds it to the Known Exploited Vulnerabilities (KEV) list. However, the vulnerability is buried among hundreds of findings in the organization’s backlog and is not remediated for three weeks. During that time, attackers exploit the flaw and establish persistence on externally facing devices.

Action at Different Maturity Levels:

- **Level 1:** There is no correlation between threat intelligence and vulnerabilities. The issue receives the same treatment as all other critical findings, despite active exploitation.
- **Level 2:** The team manually references a few public advisories but lacks the time and tools to cross-match them against local systems.
- **Level 3:** The vulnerability is matched against known exploit databases weekly. Analysts tag it for expedited remediation based on the KEV listing.
- **Level 4:** Exploitability scores (e.g., EPSS, KEV, threat feeds) are automatically ingested and influence ticketing priority and SLA timelines.
- **Level 5:** Threat data is continuously correlated with local asset exposure. Systems running the affected VPN software are automatically flagged, prioritized, and escalated to leadership. Patching occurs within 24 hours.

Outcome:

As maturity increases, the organization gains the ability to react in real time to changes in threat landscape, transforming vulnerability management from reactive backlog reduction into a threat-aligned, risk-informed process. This greatly reduces exposure windows to active threats.

Communicate

Use Case: Lack of Escalation for Repeated SLA Violations

Domain: Governance & Escalation Reporting

Scenario:

Several teams consistently fail to remediate high-risk vulnerabilities within defined SLAs. Despite repeated delays, there is no formal escalation process in place. Leadership is unaware of the growing backlog until an audit reveals non-compliance across critical systems.

Action at Different Maturity Levels:

- **Level 1:** There is no method to flag overdue remediation tasks. Missed SLAs go unnoticed unless investigated manually.\n
- **Level 2:** Escalations are sent via email when issues persist, but there is no formal process or tracking.\n
- **Level 3:** SLA breaches are monitored, and a defined escalation path notifies managers and system owners.\n
- **Level 4:** Dashboards highlight overdue items and recurring violations. Governance bodies receive regular updates and hold teams accountable.\n
- **Level 5:** Escalation metrics drive resource allocation and strategic investment. Patterns of delay trigger root cause reviews and are linked to enterprise risk reporting.

Outcome:

Organizations with mature escalation frameworks prevent long-standing risks from being buried. Visibility leads to better accountability, faster resolution, and stronger alignment with enterprise risk tolerance.

Use Case: Miscommunication Around Risk Acceptance Decisions

Domain: Exception & Risk Acceptance Communication

Scenario:

A business unit chooses not to patch a high-severity vulnerability in a legacy system, citing operational risk. The security team grants an exception, but the acceptance is not documented or shared. Months later, a penetration test exploits the unpatched flaw, catching both teams off guard. The business owner assumed the issue had been resolved.

Action at Different Maturity Levels:

- **Level 1:** Risk acceptance is handled informally. No records or communications are retained.\n
- **Level 2:** Security logs the exception in a spreadsheet, but the business unit is not copied, and the context is lost over time.\n
- **Level 3:** Accepted risks are documented in a shared portal. Notifications are sent to business owners and tracked.\n
- **Level 4:** Risk acceptance decisions include impact statements and expiration dates. Regular reports go to IT and business leadership.\n
- **Level 5:** Exceptions are fully integrated into governance dashboards and risk registers. Business, security, and IT teams all receive updates and review active exceptions quarterly.

Outcome:

As maturity increases, risk decisions become transparent, defensible, and time-bound. This fosters collaboration between business and security teams and ensures risks are not silently carried forward.

Treat

Use Case: Delayed Patching Due to Lack of Change Management Integration

Domain: Change Management Integration

Scenario:

A critical Windows vulnerability is discovered in core infrastructure systems. The remediation is straightforward but requires a scheduled reboot. The patch is delayed for over a month because the change advisory board (CAB) was not engaged early, and the remediation process was not formally linked to change controls. This gap leads to a prolonged exposure window and audit findings.

Action at Different Maturity Levels:

- **Level 1:** Remediation actions occur outside any change process. Outages are unplanned, and failed changes go unreviewed.\n
- **Level 2:** Teams notify the CAB ad hoc, but there's no formal process to prioritize security changes.\n
- **Level 3:** Security-driven changes are logged in the change system, with risk justification and urgency clearly noted.\n
- **Level 4:** Change windows are aligned with vulnerability SLAs. Security risk feeds influence CAB scheduling.\n
- **Level 5:** Vulnerability data feeds into the change system automatically. Critical patches are pre-approved based on policy and are tracked from submission to validation.

Outcome:

As maturity increases, security and IT operations become synchronized, reducing remediation delays and minimizing friction. Changes are planned, tracked, and aligned with business risk.

Use Case: Missing Closure Validation Creates False Sense of Security

Domain: Remediation Validation & Closure

Scenario:

After a ransomware scare, the organization accelerates patching across all affected systems. A follow-up scan a month later shows many systems are still vulnerable due to installation failures and reversion by rollback scripts. Because closure was assumed at the time of patch deployment, no one verified effectiveness.

Action at Different Maturity Levels:

- **Level 1:** Remediation status is updated based on ticket closure alone. There is no technical verification.\n
- **Level 2:** Some teams rescan after patching, but it is not required. Gaps are caught late, if at all.\n
- **Level 3:** Post-remediation scans are mandatory for closure. Tickets are reopened if vulnerabilities persist.\n
- **Level 4:** Scan results, log entries, or system attestations are required for closure. SLAs include validation timeframes.\n
- **Level 5:** Closure is continuously monitored. Dashboards show validation success rates, remediation cycles, and improvement trends.

Outcome:

Organizations with mature validation processes ensure remediation actually occurs as planned. This improves confidence in reporting and limits exposure from assumed (but ineffective) fixes.