UNIT-III
SECURITY RISK MANAGEMENT

RiskManagementLifeCycle–RiskProfiling–RiskExposureFactors–RiskEvaluationandMitigation– Risk Assessment Techniques–Threat and Vulnerability Management.

Risk Management:

- ✔ Risk management is a crucial aspect of Enterprise Security and Systems (ESSS). ESSS encompasses the strategies, processes, and technologies used to protect an organization's information, assets, and systems from various threats, including cyber attacks, data breaches, natural disasters, and other disruptions.
- ✔ Effective risk management in ESSS involves identifying, assessing, mitigating, and monitoring risks to ensure the security and continuity of business operations.

- ✔ Here are some key principles and steps for risk management in ESSS:

1. **Risk Identification**:
   - Identify potential risks to the organization's information, assets, and systems. This includes both internal and external threats.
   - Categorize risks into different types, such as cyber security risks, operational risks, compliance risks, and strategic risks.

2. **Risk Assessment**:
    - Evaluate the likelihood and potential impact of each identified risk. Use risk assessment methods and tools to quantify and prioritize risks.
    - Consider the organization's tolerance for risk and its business objectives when assessing risks.
3. **Risk Mitigation**:
    - Develop and implement strategies to reduce or mitigate identified risks. These strategies may include technical controls, policies, procedures, and security awareness training.
    - Allocate resources and budget for risk mitigation efforts.
4. **Incident Response Planning**:
    - Develop an incident response plan that outlines how the organization will respond to security incidents and breaches.
    - Ensure that the plan is regularly updated and that key personnel are trained on their roles and responsibilities during an incident.
5. **Security Controls**:
    - Implement a robust set of security controls and measures to protect against various types of threats. This may include firewalls, intrusion detection systems, encryption, access controls, and security patches.
    - Regularly update and test these security controls to ensure their effectiveness.

6. **Business Continuity and Disaster Recovery**:
    Establish a business continuity and disaster recovery plan to ensure the organization can continue its operations in the event of a major disruption.
    - Conduct regular backup and recovery tests to verify the plan's effectiveness.
7. **Compliance and Regulations**:
    - Stay compliant with relevant laws, regulations, and industry standards related to data security and privacy.
    - Regularly audit and assess compliance to identify and address any gaps.
8. **Monitoring and Detection**:
    - Implement continuous monitoring solutions to detect and respond to security incidents in real-time.
    - Use threat intelligence and analytics to identify emerging threats and vulnerabilities.
9. **Employee Training and Awareness**:
    - Educate employees about security best practices and raise awareness about the importance of security.
    - Conduct phishing awareness training to reduce the risk of social engineering attacks.
10. **Risk Review and Improvement**:
    - Regularly review and update the risk management strategy based on changing threats, technology advancements, and business needs.

- Conduct post-incident reviews to learn from security incidents and improve the security posture.

11. **Third-Party Risk Management**:
    - Assess and manage the security risks associated with third-party vendors and partners that have access to your systems or data.
12. **Documentation and Reporting**:
    - Maintain thorough records of risk assessments, mitigation efforts, and incident response activities.
    - Report on security metrics and key performance indicators to senior management and stakeholders.

The Risk Management Lifecycle:

STAGES OF THE RISK MANAGEMENT LIFECYCLE:

✔ Risk management is an ongoing process that involves several stages or phases, often referred to as the risk management life cycle.

✔ These stages are designed to help organizations identify, assess, mitigate, monitor, and adapt to risks effectively.

✔ The risk management process is made up of several point-in-time assessments of risk that need to be re-evaluated as risks evolve.

✔ The process begins by profiling your resources (assets) and rating them on a sensitivity scale similar to a traditional

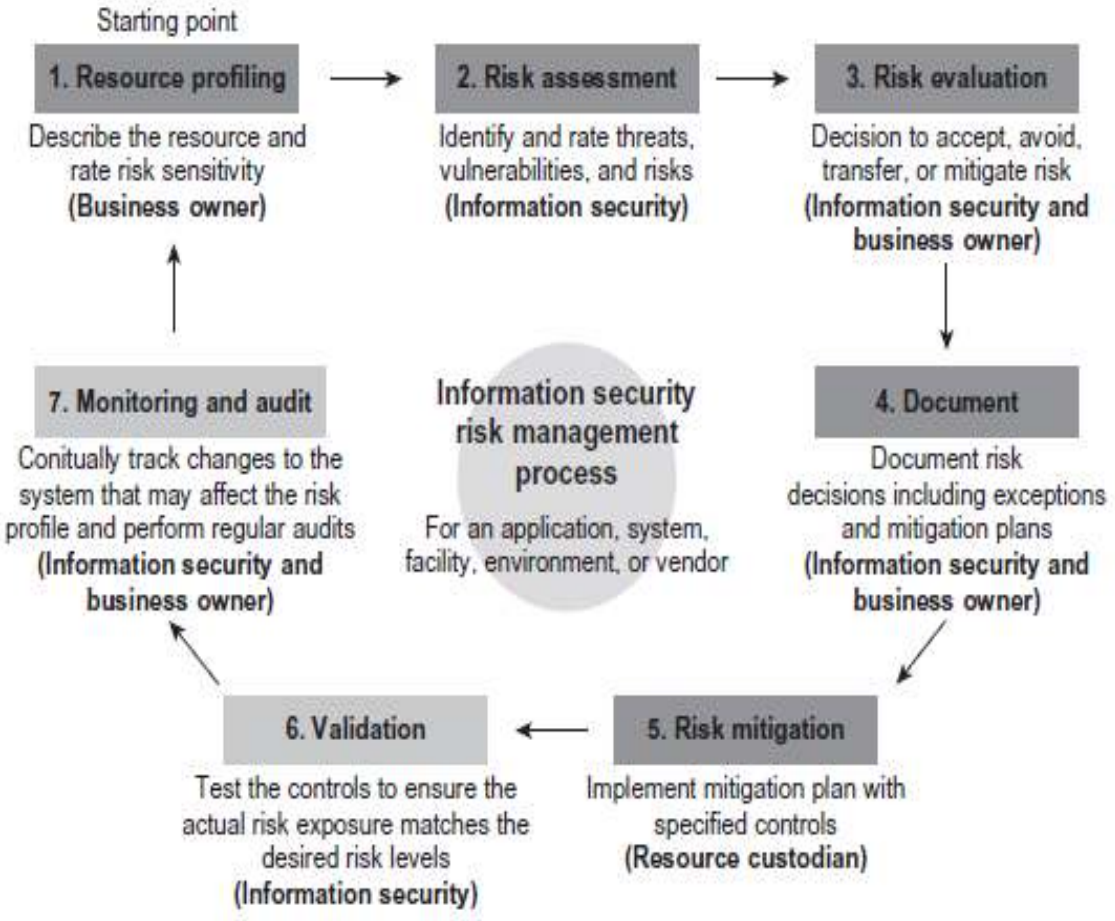
✔ Business Impact Assessment (BIA) exercise. The goal is to identify critical resources that need to be protected.

There are many times when a re-evaluation is needed (basically starting the assessment cycle over) including the following:

• A change in the sensitivity of the target resource
• A significant shift in the threat landscape
• A change in legal/regulatory requirements
• A change in security policy

STAGES OF THE RISK MANAGEMENT LIFECYCLE:

Risk Management Workflow:



FUNCTIONS:

- ✔ This process continues for the lifetime of the resource. Each stage inthe process includes a short description along with the responsible party for that step in the process
- ✔ Responsibilities for many of these steps are shared between the information security team and the business owner because the security function can't be the sole team managing risk for the organization.
- ✔ The security team can guide the process, provide oversight, and make recommendations, but ultimately it is the business that owns the risk.
- ✔ Risk assessment is the function of identifying the threats and vulnerabilities for a given resource, articulating the risk, and rating that risk exposure on a given scale.
- ✔ In this workflow, the risk assessment step includes Risk Analysis, which is considered the process of measuring (or rating) the likelihood of the undesirable event occurring and the expected severity of that event.

✔ Some kind of vulnerability assessment is generally used as part of the risk assessment to identify technical weaknesses, along with threat modeling.
✔ The next stage of risk evaluation is the function of determining the proper steps to manage that risk, whether they be to accept, mitigate, transfer, or avoid the risk exposure.

The risk management life cycle typically consists of the following phases:

Resource Profiling

- Resource profiling is a technique used in computer science and software engineering to analyze and measure the resource consumption of a program or system.
- It involves monitoring and collecting data about various system resources such as CPU usage, memory usage, disk I/O, network I/O, and more, with the goal of understanding how a program or system utilizes these resources during its execution.
- Resource profiling helps identify performance bottlenecks, memory leaks, and other resource-related issues in software applications.

Here are some key aspects of resource profiling:

1. **CPU Profiling**
2. **Memory Profiling**
3. **Disk I/O Profiling**
4. **Network I/O Profiling**
5. **Thread and Concurrency Profiling**.
6. **Power Profiling**

Risk Identification:

- In this initial phase, organizations identify potential risks and threats that could impact their objectives, projects, or operations. This includes both internal and external risks.
- Various techniques, such as brainstorming, interviews, documentation reviews, and historical data analysis, can be used to identify risks.

Risk Assessment:

- Once risks are identified, they need to be assessed to understand their likelihood and potential impact. This phase involves quantitative or qualitative analysis to prioritize risks.
- Risk assessment methods, such as risk matrices, risk heat maps, or risk scoring, can help determine which risks require further attention.

- **Risk Mitigation**:
  - In this phase, organizations develop and implement strategies to reduce or mitigate the identified risks. This may involve the implementation of controls, policies, procedures, or risk transfer mechanisms (e.g., insurance).
  - Risk mitigation strategies should align with the organization's risk tolerance and objectives.

- **Risk Monitoring and Control**:
  - After implementing risk mitigation measures, continuous monitoring is essential to ensure the effectiveness of these measures and to detect any changes in the risk landscape.
  - Monitoring may involve ongoing risk assessments, tracking key risk indicators, and using technology like risk management software.

- **Risk Reporting and Communication**:
  - Timely and clear communication of risk information is crucial to ensure that stakeholders, including senior management and relevant teams, are aware of the current risk landscape.
  - Regular reports and updates should provide insights into risk status, progress in mitigation efforts, and emerging risks.

- **Risk Review and Evaluation**:
  - Periodically, organizations should review and evaluate the effectiveness of their risk management program. This includes assessing whether mitigation measures are achieving the desired outcomes.
  - Risk assessments may need to be updated to reflect changes in the organization or external environment.

- **Risk Response and Adaptation**:
  - Based on the evaluation results, organizations may need to adjust their risk management strategies. This can involve revising mitigation measures, developing new ones, or reallocating resources.
  - Risk adaptation is essential to respond to changing circumstances and evolving risks.

- **Documentation and Record-Keeping**:
  - Maintain thorough records of all risk management activities, including risk assessments, mitigation plans, monitoring reports, and communication logs.

- Documentation helps ensure transparency, accountability, and compliance with regulatory requirements.

 Continuous Improvement:
  - The risk management life cycle is iterative and should continuously improve over time. Organizations should learn from past experiences, incidents, and changes in the risk landscape to enhance their risk management practices.

 Integration with Decision-Making:
  - Embed risk management into the organization's decision-making processes to ensure that risks are considered when making strategic and operational choices.

Risk profiling

✔ A risk profile is an evaluation of an individual's willingness and ability to take risks.
✔ A risk profile is important for determining a proper investment asset allocation for a portfolio.
✔ Organizations use a risk profile as a way to mitigate potential risks and threats.

✔ Risk profiling is a process used in financial services and investment management to assess an individual's or organization's risk tolerance, preferences, and capacity for taking on financial risk.
✔ It is an essential step in making informed decisions regarding investment portfolios, asset allocation, and financial planning.
✔ Risk profiling helps align investment strategies with the investor's or organization's comfort level for risk, ensuring that investments are suitable and in line with their financial goals and objectives.

1. **Risk Tolerance Assessment**:
   - This is the central element of risk profiling. It involves evaluating an individual's or organization's willingness and ability to accept various levels of financial risk.
   - Factors influencing risk tolerance include financial goals, investment horizon, income level, liquidity needs, and psychological factors like risk aversion.
2. **Risk Capacity Assessment**:
   - Risk capacity refers to the financial capacity to absorb losses without jeopardizing financial goals. It takes into account factors like net worth, income stability, and liquidity.
   - A person or organization with a higher risk capacity may be able to take on more significant financial risks.
3. **Risk Preferences Evaluation**:

- Risk preferences involve personal or organizational attitudes and preferences toward risk. People or entities may have different psychological inclinations, such as risk-averse, risk-neutral, or risk-seeking.
- Understanding these preferences helps tailor investment strategies to align with the investor's comfort level.

4. **Risk Profiling Questionnaires**:
    - Risk profiling often involves the use of questionnaires or surveys to gather information about an individual's or organization's risk tolerance and preferences.
    - These questionnaires ask respondents to rate their comfort level with various hypothetical risk scenarios and investment choices.
5. **Quantitative Analysis**:
    - Some risk profiling tools use quantitative metrics and financial models to assess risk tolerance and capacity more objectively.
    - For example, a quantitative approach may consider factors like standard deviation, portfolio volatility, and expected returns to recommend suitable investments.
6. **Portfolio Allocation**:
    - Based on the results of the risk profiling assessment, financial advisors or portfolio managers can recommend asset allocation strategies that align with the investor's or organization's risk profile.
    - These strategies may involve a mix of different asset classes, such as stocks, bonds, cash, and alternative investments.

7. **Regular Review and Reassessment**:
    - Risk profiles can change over time due to various factors, including changes in financial circumstances, investment goals, or market conditions.
    - It's crucial to periodically review and reassess an individual's or organization's risk profile to ensure that investment strategies remain appropriate.
8. **Education and Communication**:
    - Effective risk profiling involves educating the investor or organization about the potential risks and rewards associated with different investment options.
    - Clear communication ensures that the investor understands the implications of their chosen risk profile and investment decisions.
9. **Legal and Regulatory Compliance**:
    - Financial institutions and advisors must comply with legal and regulatory requirements when conducting risk profiling assessments.
    - Compliance ensures that recommendations are suitable and aligned with the investor's profile.

Risk Exposure:

- Risk exposure in engineering secure software systems refers to the various vulnerabilities, threats, and potential security issues that can affect a software application or system during its development, deployment, and operation phases.
- Identifying and mitigating these risks is crucial to ensuring that the software system remains secure and resilient against security threats and attacks. Here are some common risk exposure factors in engineering secure software systems:

1. **Vulnerabilities in Code:** Exposure to risks related to coding vulnerabilities, such as buffer overflows, SQL injection, and cross-site scripting (XSS), can lead to security breaches if not addressed during development.
2. **Inadequate Authentication and Authorization:** Insufficient or flawed authentication and authorization mechanisms can expose the software to unauthorized access, data breaches, or privilege escalation.
3. **Data Security Risks:** Risks related to the confidentiality, integrity, and availability of sensitive data, including personally identifiable information (PII), financial data, and intellectual property, must be considered.
4. **Insecure APIs and Interfaces:** Exposure to risks arising from insecure application programming interfaces (APIs) and external interfaces can allow attackers to manipulate data or gain unauthorized access.
5. **Lack of Input Validation:** Failing to properly validate and sanitize user inputs can lead to injection attacks, including SQL injection, command injection, and XML injection.
6. **Inadequate Security Testing:** A lack of rigorous security testing, including penetration testing and code reviews, can result in undetected vulnerabilities and weaknesses.
7. **Insufficient Logging and Monitoring:** Without proper logging and monitoring mechanisms, it becomes difficult to detect and respond to security incidents or suspicious activities.
8. **Third-Party Dependencies:** Risks associated with vulnerabilities in third-party libraries and components used in the software can affect the system's overall security posture.
9. **Weak Configuration Management:** Inadequate configuration management practices can lead to misconfigurations that expose the system to security risks.
10. **Social Engineering and Phishing:** Risks associated with social engineering attacks, such as phishing, can compromise user credentials and access to the system.
11. **Inadequate Patch Management:** Delayed or inadequate patching of known vulnerabilities can increase the risk of exploitation.
12. **Zero-Day Vulnerabilities:** Exposure to risks from undisclosed or zero-day vulnerabilities can be challenging to mitigate, as patches and mitigations may not be available.

13. **Compliance and Regulatory Risks:** Non-compliance with industry-specific regulations and data protection laws can result in legal and financial consequences.
14. **Insider Threats:** Risks associated with malicious or unintentional actions of employees or insiders can pose security challenges.
15. **Supply Chain Risks:** Risks in the software supply chain, including third-party vendors and open-source components, can introduce vulnerabilities if not carefully managed.
16. **Distributed Denial of Service (DDoS) Attacks:** Exposure to DDoS attacks can disrupt service availability and impact user experience.

FACTORS:

- Risk exposure factors in engineering secure software systems refer to the elements or variables that can influence the level of risk associated with the development, deployment, and maintenance of software with a focus on security.
- These factors help organizations assess and manage the potential security risks that may affect their software systems. Here are some key risk exposure factors in engineering secure software systems:

1. **Application Complexity:** The complexity of the software system, including its architecture, design, and codebase, can increase the risk of security vulnerabilities and make it more challenging to identify and address them.
2. **Development Practices:** Exposure to risk can be influenced by the software development methodologies, practices, and standards followed by the development team. Secure development practices and adherence to coding standards can reduce risk exposure.
3. **Software Dependencies:** Risks associated with third-party libraries, frameworks, and components used in the software can affect security. Vulnerabilities in these dependencies can lead to security issues in the software system.
4. **Security Training and Awareness:** The level of security training and awareness among development and IT staff can impact the identification and mitigation of security risks. Well-trained teams are better equipped to address security concerns.
5. **Threat Landscape:** The evolving threat landscape, including emerging cyber threats and attack techniques, can change the risk exposure over time. Staying informed about current threats is essential for managing risk.
6. **Data Sensitivity:** The type of data processed or stored by the software, such as personal information or sensitive business data, affects the potential impact of a security breach and the level of risk exposure.
7. **Authentication and Access Control:** Risks related to user authentication, access control mechanisms, and user privileges can lead to unauthorized access and data breaches.
8. **Code Quality:** The quality of the code, including the presence of vulnerabilities and coding errors, influences the risk of exploitation by attackers.

9. **Security Testing:** The extent to which security testing, such as penetration testing and code review, is conducted can affect risk exposure. Regular and thorough testing helps identify and address vulnerabilities.
10. **Incident Response Planning:** The readiness of the organization to respond to security incidents, including the existence of an incident response plan and trained incident response teams, can mitigate the impact of security breaches.
11. **Compliance Requirements:** The need to comply with industry-specific regulations and data protection laws can introduce legal and regulatory risks if not followed properly.
12. **Integration and Interconnectivity:** Risks associated with integrating the software with other systems or services, as well as the level of interconnectivity with external entities, can impact the security posture of the system.
13. **User Behavior:** Risks related to user behavior, such as weak password management or susceptibility to phishing attacks, can affect the overall security of the software system.
14. **Patch Management:** The timeliness and effectiveness of applying security patches and updates to software components and dependencies can influence risk exposure.
15. **Supply Chain Risks:** Risks in the software supply chain, including risks associated with third-party vendors and open-source software, can introduce vulnerabilities if not carefully managed.
16. **Insider Threats:** Risks related to malicious or unintentional actions by employees or insiders can pose security challenges.
17. **Regulatory Changes:** Changes in regulatory requirements related to cyber security and data protection can impact risk exposure and compliance efforts.

Qualitative Risk Exposure

Likelihood \ Severity	High	Moderate	Low
High	High	High	Moderate
Moderate	High	Moderate	Low
Low	Moderate	Low	Low

High – Corrective action must be implemented in 30 days
Moderate – Corrective action must be implemented in 90 days
Low – Corrective action must be implemented in 1 year

- There are many benefits to using a qualitative risk matrix ,First, it is nice and simple, thereby minimizing the time it will take for your peers in other business units and functions to get comfortable using it.
- Second, it makes directing the actions of the resource owners very easy. When new vulnerability announcements come out from Microsoft, for example, you can easily determine the risk exposure and communicate that to your Windows administration team.
- That team then has very clear instructions that a High exposure needs to be remediated in 30 days, and applying a patch for a Moderate exposure can take up to 90 days. Even within this structure of assigning remediation targets to each risk level, you include flexibility in your process to allow for appropriate risk acceptances to be requested and approved.
- You can even use this risk-mapping table to drive the exception approval workflow.
- For example, it may require C-level sign-off to accept a high-level risk exposure without any further mitigation. In contrast, a lower level exposure risk may be acceptable as it is or only require a manager to accept the risk.
- These mapping tables can therefore be a useful mechanism for quick and consistent risk decisions based on exposure levels

Risk Evaluation and Mitigation:

RISK EVALUATION:

Risk evaluation is a critical step in the process of engineering a secure software system. It involves assessing and analyzing potential risks to determine their impact and likelihood, allowing you to prioritize and address them effectively. Here's how you can perform risk evaluation in the context of engineering secure software systems:

1. **Identify Risks:**
    - Begin by identifying potential risks that could affect the security of your software system. These risks can include threats, vulnerabilities, and issues related to compliance and security best practices.
2. **Categorize Risks:**
    - Categorize risks based on their nature, such as technical vulnerabilities, operational risks, or compliance-related concerns. This categorization can help in organizing and addressing risks more effectively.
3. **Impact Assessment:**
    - Determine the potential impact of each identified risk. Consider how a risk could affect the confidentiality, integrity, and availability of the software and its associated data.
4. **Likelihood Assessment:**

- Evaluate the likelihood or probability of each risk occurring. Factors to consider include the complexity of the system, historical data on similar risks, and the effectiveness of existing security controls.

5. **Risk Severity Calculation:**
   - Calculate the risk severity by multiplying the impact and likelihood scores. This results in a risk rating that helps prioritize risks. Common scales include low, medium, and high severity.
6. **Risk Prioritization:**
   - Prioritize risks based on their severity scores. High-severity risks should be addressed with the highest priority, followed by medium and low-severity risks.
7. **Risk Mitigation Strategies:**
   - For each identified risk, develop and document specific mitigation strategies. These strategies should outline how the risk will be addressed, reduced, or eliminated.
8. **Cost-Benefit Analysis:**
   - Assess the cost and resources required to implement each risk mitigation strategy. Consider the potential impact of the mitigation on the overall project timeline and budget.
9. **Risk Acceptance or Transfer:**
   - Some risks may be deemed acceptable or may not justify the cost of mitigation. In such cases, document the decision to accept the risk or explore options for risk transfer, such as insurance.
10. **Continuous Monitoring:**
    - Implement continuous monitoring and review processes to track the effectiveness of risk mitigation efforts and to detect new risks that may emerge during the software development lifecycle.
11. **Documentation:**
    - Maintain detailed records of the risk evaluation process, including risk assessments, severity ratings, mitigation plans, and progress reports.
12. **Communication:**
    - Communicate the results of the risk evaluation to stakeholders, including developers, project managers, and business leaders. Ensure that everyone understands the identified risks and the plan for addressing them.
13. **Regular Updates:**
    - Periodically revisit and update the risk evaluation as the software project progresses, requirements change, or new threats emerge.

There are several options for addressing a risk:

Avoid

- This option is probably the least frequently used approach; however, it is important to keep it in mind as an option.

- Avoidance basically involves ceasing the activity that is presenting the risk altogether (or never engaging in the activity at all).
- So, if it is a new business venture or maybe a technology deployment, avoidance would be abandoning those efforts entirely.

Accept

Many risks may be unavoidable or just not worth mitigating for the organization, so in this case, management needs to make a formal decision to accept the risk.
Many organizations choose to ignore certain risks, which is really just an implicit form of acceptance.

Mitigate

- Most commonly, mitigation of a risk or remediation of a vulnerability is associated with risk management; however, remember that this is just one option.
- To mitigate a risk really means to limit the exposure in some way. This could include reducing the likelihood of occurrence, decreasing the severity of the impact, or even reducing the sensitivity of the resource. Mitigation does not imply a complete elimination of risk, just a reduction to an acceptable level.

Transfer

- This option is gaining in popularity as organizations start to really understand where the responsibilities for risks lie. The classic example of this approach is purchasing insurance to cover the expected consequences of a risk exposure.
- Data breach insurance is just starting to emerge as an option for organizations, the idea being that you transfer the risk to the insurance company.
- Risk can also be transferred through contracts with partners and clients or by pushing functions out to the customer.

RISK MITIGATION:

Mitigating risks in engineering secure software systems is crucial to ensuring the confidentiality, integrity, and availability of the software and the data it handles.

Three general categories of risk mitigation are as follows:

- Risk Alleviation – implements controls to prevent the threat/vulnerability (such as patching a software weakness).
- Risk Limitation – limits likelihood or effects with controls (such as the examples given above).
- Risk Planning – develops a formal plan to prioritize, implement, and maintain controls (this doesn't directly change the risk exposure level, but it assumes some plan to address the risk in the near future, therefore, limiting the time frame for possible exposure).

POLICY EXCEPTIONS AND RISK ACCEPTANCE

Risk exceptions serve many purposes, including documenting when

- a risk exposure can't be remediated or mitigated
- the business chooses to accept a risk as is
- the cost of mitigation outweighs the impact of the exposure.
- compensating controls already exist
- a risk exposure needs to be accepted temporarily while it is being addressed.

Exception Workflow

One possible exception request and approval workflow can be implemented as follows:

1. Anyone can submit an exception request.
2. Someone reviews the exception for their functional area, selects the appropriate member of management to approve for that functional unit, and forwards it to the information security team.
3. Information security team reviews and enters a risk rating and expiration date,then forwards to the member of management to approve the exception request for the affected functional area.
4. Depending on the level of risk, senior management reviews and approves the exception request and/or the mitigation plan.
5. Depending on the risk level, the head of information security or a combination of corporate officers approves the exception request.
6. Exception is finalized, and any actions are tracked until it expires.
7. Information security team reviews updates from requestor and either closes the exception or extends it.

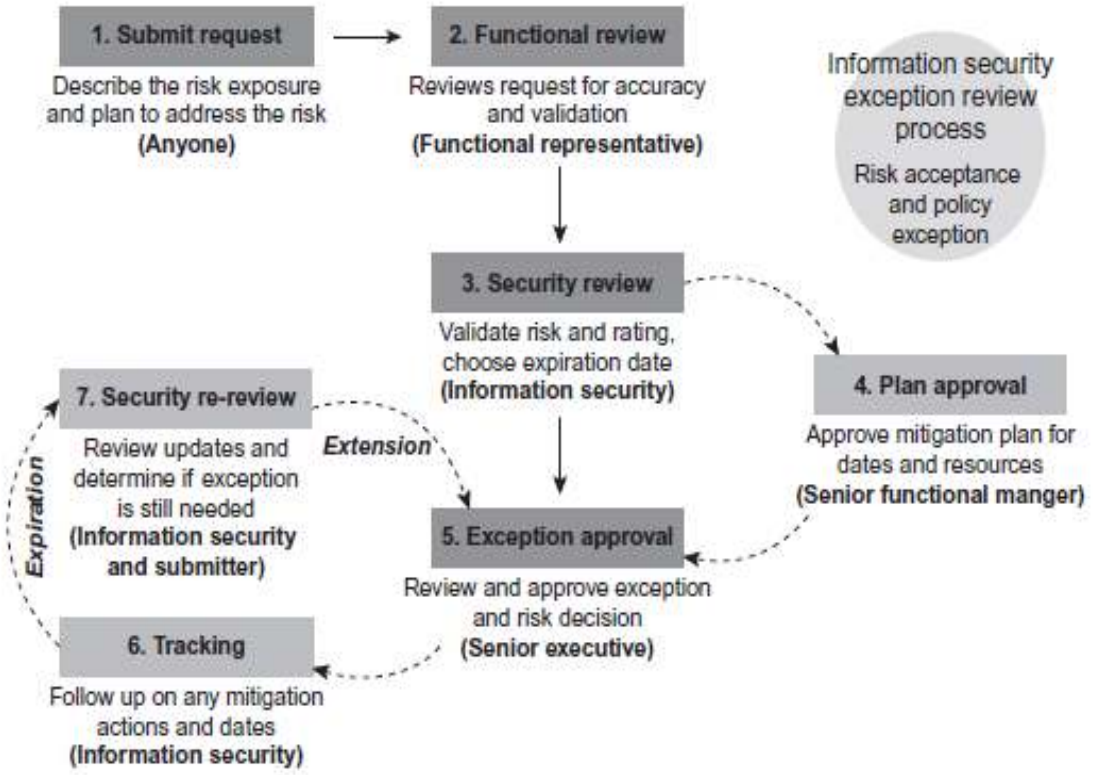


FIGURE 8.1

Risk exception approval workflow.

Risk Assessment Techniques

It involves identifying, analyzing, and evaluating potential security risks and vulnerabilities that could impact the confidentiality, integrity, and availability of the software and the data it handles.

TECHNIQUES:

- Operational Assessments
- Project-Based Assessments
- Third-Party Assessments

Operational Assessments

- The operational assessments will encompass regular assessments of emerging threats, newly announced vulnerabilities, and discovered standard violations.

- Operational assessments should not be confused with assessments of risks in the operations domain.
- In contrast, an assessment of the operations domain would define the scope of the assessment, which would focus on threats to operations continuity.

Some examples of operational risk assessment tasks in the information security space include the following:

- Threat analysis
- Vulnerability scanning
- Patch remediation
- Penetration Testing
- Incident prioritization
- Exception processing
- Compliance to standards reviews
- Certification and accreditation (C&A)
- Auditing (internal or external)
- Responses to client due diligence evaluations
- Vendor on-site reviews
- Regulatory gap analysis

Operational Techniques

For all those potential operational assessments, your options really come down to just a few assessment formats:

- **Questionnaire**
- **Interview**
- **Passive testing**
- **Active testing**
- **Review of third-party assessment**
- **Acceptance of a certification**

Types of assessments:

- Enterprise vulnerability assessment (active)
- Penetration testing analysis (active)
- Wireless security assessment (active)
- Black box application testing (active)
- Malicious threat assessment (passive)

• Internet reconnaissance (passive)
• Application code security review (passive)

PROJECT-BASED ASSESSMENTS

Each requires a slightly different approach and has its own challenges.
• Software development
• Software/technology acquisition
• Selection of third-party service provider

The scope of an assessment can vary greatly, from a new product enhancement to the acquisition of another company.

Risk Assessments in the Project Lifecycle:

- A security risk assessment can be performed by just about any one involved in the project team if given the proper guidelines, and occasionally, the project may require an outside party to guide the assessment.
- Your organization's culture will strongly influence who should lead each assessment, but generally the responsibility will fall on the Information Security team.
- The output of this assessment will include the identification of risks, threats, and general concerns from the team and, ultimately, recommendations for controls to mitigate those threats.
- The analysis and recommendations would then generally be presented to senior management or other project stakeholders to make the final decisions.

Third-Party Reviews and Certifications

- When working with vendors and service providers, you are going to need to rely on other means of assessing the security posture of the third party.
- Most service providers aren't going to let you show up at their offices with a security scanner and just let you go nuts on their environment (at least we hope they won't!).
- Thus begins the negotiation of best evidence. You might think of this as a similar dilemma to what you would see in court. Direct evidence may not always be available, so you may need to rely on alternatives like maybe an expert witness.

- The same is often true when assessing a third-party provider—you may not be allowed to walk through their Security Operations Center (SOC) or run your own penetration test against their Internet-facing systems.

Baseline Reviews:

- C&A process is meant to formalize the standards for configuring a system securely and force an explicit review of those controls and authorization decision to allow it to operate in an environment.
- Certification and accreditation are really both subsets of an overall information security risk management program.
- Risk management is the overall program for identifying weaknesses, threats to those weaknesses, and assessing the impact to the organization that might result from an exploitation of those weaknesses.
- Certification is the process of evaluating whether the system/application meets the minimum standards that have been established, and accreditation is the management decision process to determine if any deviations from standards are acceptable.

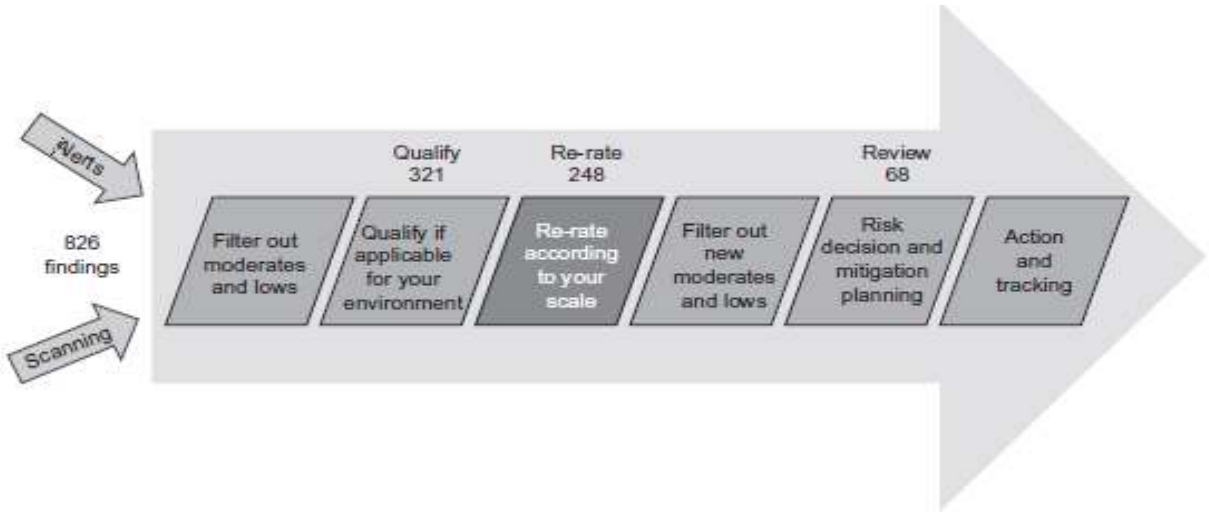
There are two contexts in which the term “baseline” is used for Information
Security.

- The first is referring to a point in time snapshot of the current state of the environment as a comparison point.
- The second is the minimum set of required configuration settings or controls to meet a desired level of security.

Threat and Vulnerability Management:

Threat and Vulnerability Management (TVM) is a crucial part of the software development lifecycle, especially in engineering secure software systems. Secure software development involves identifying, mitigating, and managing threats and vulnerabilities throughout the development process.

Mechanism:



- Information Security team is responsible, many of the operational activities will include processing of new vulnerability notifications, keeping up to date on the news and reports of the latest emerging threats, scanning the environment for vulnerabilities and unauthorized services, reviewing the results of penetration tests, and working with the operational teams to ensure that security patches are being rolled out.
- All of these activities can be grouped under the umbrella of a Threat and Vulnerability Management (TVM) program. This program is also often referred to as Threat and Vulnerability Assessment (TVA), but TVM is more accurate because the process needs to include more than just identifying and ranking the risk exposures.

Consider the following scenario where

1. a vendor announces a new critical vulnerability in their Web application server software affecting a particular service;
2. you then read on the SANS Internet Storm Center diary that there have been several confirmed exploits of that vulnerability in the wild;
3. you look through your software inventory and see that you run the affected Version on all your Web servers;
4. you consult the most recent scan of your Internet presence and find that the particular vulnerable service is indeed running on your Web servers;
5. you confirm several failed attempts to exploit this service by consulting the Web server logs from your central log management system .

6. and, finally, you coordinate with your server management team to roll out the vendor patch that evening during a maintenance window.

Program Essentials:

Start with the following TVM development steps:

1. Establish an asset inventory

• System Type and Version
• Software (including Version)
• Physical and Logical Location
• Logical Network Addressing
• Owner
• Resource Administrator
• Data Sensitivity

2. Profile your environments (sensitivity)

• General Description
• Function and Features
• Information Classification
• Criticality to Organization
• Applicable Regulations
• User Community

3. Define your risk scales

Table 11.1 Asset Class by Sensitivity

Asset Class	Sensitivity
Production servers	High
Desktop/Laptop	Moderate
Printers	Low
Infrastructure	High

- **Requirements Analysis**:
    - Start by defining security requirements during the initial stages of the software development process. This includes identifying

potential threats and vulnerabilities that the software system may face.

- **Secure Design**:
  - Implement secure design principles to minimize vulnerabilities from the ground up. Consider security patterns and architecture that help protect against common threats like SQL injection, cross-site scripting (XSS), and authentication issues.
- **Threat Modeling**:
  - Conduct threat modeling exercises to systematically identify and assess potential threats and vulnerabilities in the software's architecture and design. Tools like STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, Elevation of privilege) can be helpful.
- **Code Review**:
  - Perform regular code reviews with a focus on security. Static code analysis tools can help automate the detection of security vulnerabilities in the codebase.
- **Penetration Testing**:
  - Conduct penetration testing and security assessments to simulate real-world attacks and discover vulnerabilities that may not be apparent through automated testing.
- **Secure Coding Practices**:
  - Enforce secure coding practices among developers, such as input validation, output encoding, and proper error handling, to prevent common vulnerabilities like injection attacks and buffer overflows.

- **Dependency Scanning**:
  - Regularly scan and update third-party libraries and components to ensure they are free from known vulnerabilities. Tools like OWASP Dependency-Check can help automate this process.
- **Security Training**:
  - Provide ongoing security training and awareness programs for development teams to ensure they stay up-to-date with the latest security best practices.
- **Patch Management**:
  - Maintain a process for promptly applying security patches and updates to the software and its dependencies to address newly discovered vulnerabilities.
- **Security Testing**:
  - Incorporate security testing into the continuous integration/continuous deployment (CI/CD) pipeline to automatically identify security issues during the development process.
- **Logging and Monitoring**:
  - Implement robust logging and monitoring mechanisms to detect and respond to security incidents in real-time.
- **Incident Response Plan**:

    - Develop and maintain an incident response plan that outlines procedures for identifying, mitigating, and recovering from security incidents.
- **Security Documentation**:
    - Maintain documentation that includes security-related information, such as threat models, security controls, and incident response procedures.
- **Compliance and Standards**:
    - Ensure compliance with industry-specific security standards and regulations (e.g., OWASP Top Ten, ISO 27001, NIST) relevant to your software and industry.
- **Risk Management**:
    - Continuously assess and prioritize security risks based on the software's sensitivity, potential impact, and likelihood of exploitation. Allocate resources to address high-priority risks.
- **Security Reviews**:
    - Periodically review and update security controls and measures to adapt to evolving threats and vulnerabilities.