CISSP Process Guide
V.21

I'm Fadi Sodah (aka madunix), and I'm an IT Director. I've been in the IT realm for over twenty-six years and have held a variety of positions. I worked as a networks engineer, systems engineer and security engineer and I was among the Top 100 Hall of Fame HackTheBox. I'm an active member of Experts-Exchange (EE) since 2004. I have been awarded the Most Valuable Expert (EE MVE) in 2019. You can find me on Experts-Exchange (EE), LinkedIn, Facebook, Telegram, Discord and Twitter @madunix. I hold certifications in many areas of the IT field such as networking, systems, audit, IoT, AI and security: PCCSA, PCNSA, PCNSE, CCNP, CCIP, CISA, CISSP, CFR, CSC, ACE, CIoTSP, CAIP, CISM, eJPT, CyberSafe, SCSC, KCSP, KCTP, OCIF, OADCS, ADCI and ICATE.

To benefit others with the knowledge and experienced I gained during my study term, I have summarized the main underlying concepts in a general overview. I am hoping this consolidation of core concepts and processes would benefit those interested in becoming security experts.

This document intends to be supplementary, not a replacement for officially published study guides and books. I may have added multiple definitions of the same process or procedure due to the varying definitions from different resources such as the Official CBK, Sybex, NIST publications, SANS papers, or the AIO Shon Harris books. If you encounter any conflicts, please refer to the latest Official books CISSP CBK, AIO and Sybex. Being a CISSP candidate, you should fully understand CISSP concepts, methodologies and their implementations within the organization.

The CISSP exam is designed to test your presence of mind, knowledge, experience, concept and hardworking.

- Use Sybex as a baseline for your study
- In case of misconception keep referring to CBK CISSP book and index
- Review the notes from Sunflower powered by Nick Gill
- Review CISSP Process Guide powered by madunix
- Review Memory Palace CISSP Notes powered by Prashant
- If you study by yourself, you will always see your material from the same perspective; I recommend to choose a study group telegram and discord.
- Review NIST publication
- Check CISSP references www.isc2.org/Certifications/References
- Measure your progress through quizzes and practice exams, be aware don't go by the score try to fill your gaps
- Keep checking the (ESG) Elite Security Groups
    - https://thorteaches.com/cissp/
    - https://www.studynotesandtheory.com/
    - https://wentzwu.com/
    - https://prabhnair.in/
    - https://www.experts-exchange.com/members/madunix

Do not try any shortcut when it comes to reading books and gaining knowledge. This quick reference should be utilized as a fast recap of security concepts. It's essential that you read Official CISSP books first and then use these notes to get a recap of what you have learned. I wish you good luck for the CISSP exam.

You can send me a donation to my account to keep this document updated: paypal.me/FadiSodah
Email:madunix@gmail.com

Corporate Governance:
Corporate governance is the set of responsibilities and practices exercised by the board and executive management with the goal of providing strategic direction, ensuring that objectives are achieved, ascertaining that risk is managed appropriately and verifying that the enterprise's resources are used responsibly.
- Auditing supply chains
- Board and management structure and process
- Corporate responsibility and compliance
- Financial transparency and information disclosure
- Ownership structure and exercise of control rights

Governance, Risk and Compliance (GRC):
The process of how an organization manages its information resources. This process usually includes all aspects of how decisions are made for that organization, such as policies, roles and procedures the organization uses to make those decisions. It is designed to ensure the business focuses on core activities, clarifies who in the organization has the authority to make decisions, determines accountability for actions and responsibility for outcomes, and addresses how expected performance will be evaluated.

Areas of focus for IT Governance:
- Strategic alignment
- Value delivery
- Resource management
- Risk management
- Performance management

Governance vs. Management:
- Oversight vs. Implementation
- Assigning authority vs. authorizing actions
- Enacting policy vs. enforcing
- Accountability vs. responsibility
- Strategic planning vs. project planning
- Resource allocation vs. resource utilization

Note: Governance: (What do we need to accomplish). Governance typically focuses on the alignment of internal requirements, such as corporate policies, business objectives, and strategy. Management: (How)

Security Policy:
- Define the scope
- Identify all assets
- Determine level of protection
- Determine personal responsibility
- Develop consequences for noncompliance

Securing the Infrastructure:
- Framework for Governance
- Risk Management
- The Security Program
- Data Protection
- System and Data Management
- Security Awareness Training
- User Provisioning
- Monitoring and Enforcement
- Incident Response

The importance of following Infosec standards:
Creating and using common, proven practices is an important part of a successful information security program. Not only do standards support proactive management and efficient risk mitigation, adopting and consistently following a standard can bring additional benefits to any organization.
- TRUST & CONFIDENCE. When organizations obtain certifications that demonstrate compliance, they create a sense of trust and confidence among employees and third parties with whom they interact.
- BETTER RESULTS. When you speak the same jargon, results are more productive, effective, and cohesive. E.g., vendor assessments can be smoother and faster with a formal infosec program in place.
- COMPETITIVE ADVANTAGE. Developing a formal infosec program and obtaining certification boosts client and stakeholder confidence in how infosec risks are managed and aligned with their own risk appetite.
- CORPORATE RESPONSIBILITY. Holding an infosec certification can help organizations demonstrate due diligence and due care, which are mandatory requirements for company officers and essential for mitigating corporate negligence.

Note: Information security standards offer best practices and share expert information. These standards allow organizations to adopt, tailor, and implement a valuable infosec program without having to hire full time experts, reinventing the wheel, and learning by trial and error, which is costly, time consuming and dangerous.

Challenges of implementing and maintaining standards:
- Time: Implementing and maintaining information security standards is not a one-time project. Rather, it is a process that requires dedicated, qualified personnel, support from senior leadership, and continuous monitoring and improvement. A successful effort will require buy-in from the entire organization.
- Cost: Standards can be expensive to implement and just as costly to maintain. In the case of ISO 27001, for example, in addition to the time and effort necessary to meet the standard requirements, organizations must budget for annual audit fees, which can be substantial.
- Buy-in: Senior leadership buy-in and program ownership at the C-level are critical elements for an organization to deploy an information security program effectively. The information security team must share metrics, report the effectiveness of the program, and demonstrate its value and strategic alignment with the organization's business objectives to maintain senior leadership support.
- Change management: In general, everyone appreciates the value of securing information until it requires a change. Security teams implementing standards are challenged to strike a delicate balance between security and convenience.
- Continuous improvement: Standards have life cycles. When a standard is updated, it is the responsibility of all compliant organizations to be aware of the updates and implement them by specified dates, or as soon as possible if a time line is not mandated. In some cases, a standard might become obsolete, and a new standard must be researched and presented to senior leadership for approval for implementation.

Access Control Review:
The following is a review of the basic concepts in access control.
Identification:
• Subjects supplying identification information
• Username, user ID, account number
Authentication:
• Verifying the identification information
• Passphrase, PIN value, thumbprint, smart card, one-time password
Authorization:
• Using the identity of the subject together with other criteria to make a determination of operations that a subject can carry out on objects
• "I know who you are, now what am I going to allow you to do?"
Accountability:
• Audit logs and monitoring to track subject activities with objects

Authorization approval procedure:
• Formalized
• Approval by the direct manager, data owner, security professional
• Access permissions follow the principle of least privilege
• Balance security with the need for access
• Avoid allowing too much privilege — Conflicts of interest
• Remove privilege when no longer needed

Due Diligence vs. Due Care:
• Due Diligence: "Researching" -- Investigating and understanding risks
• Due Diligence: "Doing" all the necessary tasks required to maintain the due care
• Due Care: "Doing" -- Developing policies and procedures to address risk
• Due Care is to act responsibly

Data Protection:
When you think about data protection, there are essentially 5 key trends to be aware of:

- As always, the ability to recover data in the event of a loss or corruption is critical to why business does back up. It is a must.
- Next is disaster recovery (DR). In much the same way as application or data recovery, in the event of a natural disaster, the ability to get the business up and running is paramount. Statistically, businesses that can't recover from a disaster within 72 hours go out of business, so having a plan is critical, no matter the size of the business.
- Business continuity is a superset of DR and having a business continuity plan would mean having a good DR plan. It is imperative that not only are applications protected, but users can access the data and applications in the event of a disaster.
- The ability to reuse existing data for other business purposes. With the latest talk about "data being the new oil" or "natural useable resource," companies that can take advantage of this data are more likely to be successful. Having the ability to spin up copies of this data quickly for other business uses such as DevOps, analytics, or reporting as well as supporting a good DR strategy has become a way to take further advantage of your backup solution.
- The latest entry to the list is cyber resiliency. While cyber resiliency has been important for a long time, it is now top of everyone's mind due to the most recent attacks and the statistics that talk about how cyber attacks cost businesses a lot of money. The ability to recover from one of these attacks is not as simple as just a data recovery, so new planning has to be part of how businesses protect their data.

Data at Rest:
The term data at rest refers to data that lives in external or auxiliary storage devices,
such as hard disk drives (HDDs), solid-state drives (SSDs), optical discs (CD/DVD), or even on magnetic tape. A challenge to protect the data in these states is, it is vulnerable, not only to threat actors attempting to reach it over our systems and networks but also to anyone who can gain physical access to the device. Data protection strategies include secure access controls, the segregation of duties, and the implementation of the need to know mechanisms for sensitive data.

Data in Use:
Data in use refers to the information that is currently in use. It is used by staff, as in laptops or portable devices, and information that is being printed or copied to a USB stick. This is the data available in endpoints. Data security controls for data in use would include port protection and whole disk encryption. Controls against shoulder surfing, such as clear screen and clear desk policies, are also applicable to data in user controls.

Security:
Security is a continuous process, not a one-shot project. The security life cycle or the security wheel is a continuous process that consists of several consequent phases (stages). The word cycle indicates the continuous and endless nature of such process. The ISO 27001 defines the cycle of the information security management system ISMS as PCDA: Plan-Do-Check-Act.

Samples of testing CIA Triad:
• Security Functionality: Verify that the software behaves according to requirements, which should include security.
• Fuzz-testing (or fuzzing): Enter a wide variety of out-of-range
• Dynamic Validation: Use variable data in the code to ensure the integrity of the software.
• Risk-Based Testing: Prioritize what features to test based on their potential risk and the impact of their failure.
• Penetration Testing: Play the role of an attacker, finding weaknesses and attempting exploits.
• Authentication Testing: Verify that communication over a network such as the Internet is protected by secure identification methods.
• Regression Testing Confirm that newer patches, updates, and fixes work with older code.

Considerations for Security Controls include:
• Accountability (can be held responsible)
• Auditability (can it be tested?)
• A trusted source (source is known)
• Independence (self-determining)
• Consistently applied
• Cost-effective
• Reliable
• Independence from other security controls (no overlap)
• Ease of use
• Automation
• Sustainable
• Secure
• Protects confidentiality, integrity, and availability of assets
• Can be "backed out" in the event of an issue
• Creates no additional issues during operation
• Leaves no residual data from its function

Business Impact Assessment (BIA):
A systematic process to determine and evaluate the potential effects of an interruption to critical business operations as a result of exploitation, disaster, accident or emergency.

Business Impact Assessment:
• Identify Priorities
• Identify Risk
• Likelihood Assessment
• Impact Assessment
• Resource prioritization
Risk can never be mitigated to zero (there is no such thing as "no risk" or "perfect security")

Business Impact Analysis:
• Identify critical functions
• Identify critical resources
• Calculate MTD for resources
• Identify threats
• Calculate risks
• Identify backup solutions

Business Impact Analysis:
• Select individuals to interview for data gathering
• Create data-gathering techniques
• Identify critical business functions
• Identify resources these functions depend upon
• Calculate how long these functions can survive without these resources
• Identify vulnerabilities and threats
• Calculate the risk for each different business function
• Document findings and report them to management

Key Performance Indicator KPI based on:
• BIA
• Effort to implement
• Reliability
• Sensitivity

Security Programs Metrics:
• KPI looks backward at historical performance
• KRI looks forward, show how much risk exists that may jeopardize the future security of the organization.

Business Continuity Planning (BCP):
• Project Initiation
• Business Impact Analysis
• Recovery Strategy
• Plan design and development
• Implementation
• Testing
• Continual Maintenance

BCP (NIST 800-34):
• Develop a planning policy;
• BIA
• Identify preventive controls
• Create contingency strategies
• Develop contingency plans
• Test
• Maintenance

Business Continuity Planning (BCP):
• Provide immediate and appropriate response to emergency situations
• Protect lives and ensure safety
• Reduce business impact
• Resume critical business functions
• Work with outside vendors and partners during the recovery period
• Reduce confusion during a crisis
• Ensure survivability of the business
• Get "up and running" quickly after a disaster

DRP vs. BCP:
• BCP - Corrective Control
• DRP - Recovery Control
• Both BCP and DRP - fall under the category of Compensating Control
• BCP – is not a preventive control as it can NOT prevent a disaster
• BCP - helps in the continuity of organization function in the event of a disaster
• BCP - maintaining critical functions during a disruption of normal operations
• DRP - recovering to normal operations after a disruption

Business Continuity Planning (BCP):
• Continuity Policy
• Business Impact Assessment (BIA)
• Identify Preventive Controls
• Develop Recovery Strategies
• Develop BCP
• Exercise/Drill/Test
• Maintain BCP

DR Team:
• Rescue Team: Responsible for dealing with the immediacy of the disaster –employee evacuation, crashing the server room, etc.
• Recovery Team: Responsible for getting the alternate facility up and running and restoring the most critical services first.
• Salvage Team: Responsible for the return of operations to the original or permanent facility (reconstitution) – (get us back to the stage of normalcy)

Business Continuity Planning (BCP) Documents:
• Continuity of planning goals
• Statement of importance and statement of priorities
• Statement of Organizational responsibilities
• Statement of Urgency and Timing
• Risk assessment, Risk Acceptance, and Risk mitigation document
• Vital Records Program
• Emergency Response Guidelines
• Documentation for maintaining and testing the plan

DRP/BCP document plan should be:
• Created for an enterprise with individual functional managers responsible for plans specific to their departments
• Copies of the plan should be kept in multiple locations
• Both Electronic and paper copies should be kept
• The plan should be distributed to those with a need to know
• Most employers will only see a small portion of the plan

Business Continuity Planning (BCP):
• Project scope and planning
  •• Business Organization Analysis
  •• BCP team selection
  •• Resource Requirements
  •• Legal and regulatory requirements
• Business impact assessment
  •• Identify priorities
  •• Risk Identification
  •• Likelihood Assessment
  •• Impact Assessment
  •• Resource Prioritization
• Continuity planning
  •• Strategy Development
  •• Provisions and Processes
  •• Plan Approval
  •• Plan Implementation
  •• Training and Education
• Approval and implementation
  •• Approval by senior management (APPROVAL)
  •• Creating an awareness of the plan enterprise-wide (AWARENESS)
  •• Maintenance of the plan, including updating when needed (MAINTENANCE)
  •• Implementation

Development of Disaster Recovery Plan (DRP):
• Plan Scope and Objectives
• Business Recovery Organization (BRO) and Responsibilities (Recovery Team)
• Major Plan Components - format and structure
• Scenario to Execute Plan
• Escalation, Notification and Plan Activation
• Vital Records and Off-Site Storage Program
• Personnel Control Program
• Data Loss Limitations
• Plan Administration

Disaster Recovery Plan (DRP) procedures:
• Respond to disaster by a pre-defined disaster level
• Assess damage and estimate time required to resume operations
• Perform salvage and repair

Elements of Recovery Strategies:
• Business recovery strategy
  •• Focus on the recovery of business operations
• Facility & supply recovery strategy
  •• Focus on facility restoration and enable alternate recovery site(s)
• User recovery strategy
  •• Focus on people and accommodations
• Technical recovery strategy
  •• Focus on the recovery of IT services
• Data recovery strategy
  •• Focus on the recovery of information assets

The eight R's of a successful Recovery Plan:
• Reason for planning
• Recognition
• Reaction
• Recovery
• Restoration
• Return to Normal
• Rest and Relax
• Re-evaluate and Re-document

Disaster Recovery Program:
- Critical Application Assessment
- Backup Procedures
- Recovery Procedures
- Implementation Procedures
- Test Procedures
- Plan Maintenance

Post-Incident Review:
The purpose is how we get better; after a test or disaster has taken place:
- Focus on how to improve
- What should have happened?
- What should happen next?
- Not who´s fault it was; this is not productive

Continuity Planning:
Normally applies to the mission/business itself; Concerns the ability to continue critical functions and processes during and after an emergency event.

Contingency Planning:
Applies to information systems, and provides the steps needed to recover the operation of all or part of the designated information system at an existing or new location in an emergency.

Business Continuity Plan (BCP):
BCP focuses on sustaining an organization's mission/business process during and after a disruption. It May be used for long-term recovery in conjunction with the COOP plan, allowing for additional functions to come online as resources or time allows.

Occupant Emergency Plan (OEP):
It outlines first-response procedures for occupants of a facility in the event of a threat or incident to the health and safety of the personnel, the environment, or property.

Cyber Incident Response Planning (CIRP):
It's A type of plan that normally focuses on detection, response, and recovery to a computer security incident or event. It establishes procedures to address cyber-attacks against an organization's information system(s).

Information System Contingency Plan (ISCP):
It provides established procedures for the assessment and recovery of a system following a system disruption. Provides key information needed for system recovery, including roles and responsibilities, inventory info, assessment procedures, detailed recovery procedures, and testing of a system.

Continuity of Operations Plan (COOP):
It focuses on restoring an organization's mission essential function of an alternate site and performing those functions for up to 30 days before returning to normal operations.

Disaster Recovery Plan (DRP):
Applies to major physical disruptions to service that deny access to the primary facility infrastructure for an extended period. An information system-focused plan designed to restore operability of the target system, application, or computer facility infrastructure at an alternate site after an emergency. Only addresses information system disruptions that require relocation.

Risks to the organization found in:
- Financial
- Reputational
- Regulatory

Risk Analysis:
- Analyzing the environment for risks
- Creating a cost/benefit report for safeguards
- Evaluating threat

Elements of risk:
- Threats
- Assets
- Mitigating factors

Risk Analysis methodology:
- CRAMM (CCTA Risk Analysis and Management Method)
- FMEA (Failure modes and effect analysis methodology)
- FRAP (Facilitated Risk Analysis Process)
- OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation)
- PUSH
- Spanning Tree Analysis
- SOMAP (Security Officers Management and Analysis Project)
- VAR (Value at risk)

RMF CSIAAM: (NIST 800-37):
The risk management framework (RMF) encompasses a broad range of activities to identify, control, and mitigate risks to an information system during the system development life cycle. One of the activities is the development of an ISCP. Implementing the risk management framework can prevent or reduce the likelihood of the threats and limit the consequences of risks. RMF include:
- Categorize the information system and the data
- Select an initial set of baseline security controls
- Implement the security controls and describe how the controls are employed
- Assess the security controls
- Authorize systems to be launched
- Monitor the security controls

Risk Management Process: (FARM):
- Framing risk
- Assessing risk
- Responding to risk
- Monitoring risk

Risk management Policy Document:
- Objectives of the policy and rationale for managing risk
- Scope and charter of information risk management
- Links between the risk management policy and the organizations strategic and corporate business plans-Extent and range of issues to which the policy applies
- Guidance on what is considered acceptable risk levels
- Risk management responsibilities
- Support expertise available to assist those responsible for managing risk
- Degree of documentation required for various risk-management related activities, e.g., change management
- A plan for reviewing compliance with the risk management policy
- Incident and event severity levels
- Risk reporting and escalation procedures, format and frequency

Risk Management Life Cycle:
- Continuously monitoring
- Evaluating
- Assessing and reporting risk.

Risk management:
- Risk Assessment — Identify Assets, Threats Vulnerabilities
- Risk Analysis — Value of Potential Risk
- Risk Mitigation — Responding to Risk
- Risk Monitoring — Risk is forever

Risk management entails evaluating:
- Threats
- Vulnerabilities
- Countermeasures

Methodologies of Risk Assessment:
- Prepare for the assessment.
- Conduct the assessment:
  - Identify threat sources and events.
  - Identify vulnerabilities and predisposing conditions.
  - Determine the likelihood of occurrence.
  - Determine the magnitude of impact.
  - Determine risk.
- Communicate results.
- Maintain assessment.

Preparing Risk Assessment:
- Purpose of the assessment
- The scope of the assessment
- Assumptions and constraints associated with the assessment
- Sources of information to be used as inputs to the assessment
- Risk model and analytic approaches

Risk Assessment (NIST 800-30):
- System / Asst. Characterization
- Threat Identification
- Vulnerability Identification
- Control Analysis
- Likelihood Determination
- Impact Analysis
- Risk Determination
- Control Recommendations
- Results Documentation

Key Challenges in Third-Party Risk Management:
- Increases the complexity of third-party network & it's managnt.
- Risk of failure to manage regulatory compliances
- Additional Cost for monitoring third-parties
- Lack of collaboration among parties
- Risk of information / data leakage

Key Components of Third-Party Risk Management Framework:

Following are the key components of Third-Party Risk Management (TPRM) Framework:
- Planning & process definition
- Segmentation & Screening
- Qualification
- Security & Permissions
- Workflows
- Risk Mitigation
- Continuous Monitoring
- Reports & Dashboard
- Centralized Repository
- Alert & Notification

Damage assessment:
- Determining the cause of the disaster is the first step of the damage assessment
- How long it will take to bring critical functions back online
- Identifying the resources that must be replaced immediately
- Declare a disaster

Damage assessment:
- Determine the cause of the disaster.
- Determine the potential for further damage.
- Identify the affected business functions and areas.
- Identify the level of functionality for the critical resources.
- Identify the resources that must be replaced immediately.
- Estimate how long it will take to bring critical functions back online.
- If it will take longer than the previously estimated MTD values to restore operations, then a disaster should be declared and BCP should be put into action.

Note:
- The first activity in every recovery plan is damage assessment, immediately followed by damage mitigation.
- The final step in a damage assessment is to declare a disaster.
- The decision to activate a disaster recovery plan is made after damage assessment and evaluation is completed.

Configuration Management:
- Plan
- Approve Baseline
- Implement
- Control Changes
- Monitor
- Report
- Repeatable

Configuration Management:
- Configuration Identification
- Configuration Control
- Configuration Status Accounting
- Configuration Audit

Change Control:
- Implement changes in a monitored and orderly manner.
- Changes are always controlled
- Formalized testing
- Reversed/rollback
- Users are informed of changes before they occur to prevent loss of productivity.
- The effects of changes are systematically analyzed.
- The negative impact of changes in capabilities, functionality, performance
- Changes are reviewed and approved by a CAB (change approval board).

Change Management:
- Request for a change to take place
- Approval of the change
- Documentation of the change
- Tested and presented
- Implementation
- Report change to management

Change Management:
- Request
- Review
- Approve
- Schedule
- Document

Change Management:
- Request
- Evaluate
- Test
- Rollback
- Approve
- Document
- Determine Change Window
- Implement
- Verify
- Close

Patch Management:
- Patch Information Sources
- Prioritization
- Scheduling
- Testing
- Installation
- Assessment
- Audit
- Consistency
- Compliance

Patch Management:
- Evaluate
- Test
- Approve
- Deploy
- Verify

Patch Management:
- Inventory
- Allocate Resources
- Pursue updates
- Test
- Change Approval
- Deployment plan
- Rollback plan
- Deploy and verify the updates with policy requirements
- Document

Problem Management:
- Incident notification
- Root cause analysis
- Solution determination
- Request for change
- Implement solution
- Monitor/report

Information Systems Security Engineering (ISSE) Process:
- Discover Information Protection Needs; ascertain the system purpose.
- Identify information asset needs protection.
- Define System Security Requirements; Define requirements based on the protection needs.
- Design System Security Architecture; Design system architecture to meet security requirements.
- Develop Detailed Security Design; Based on security architecture, design security functions and features of the system.
- Implement System Security; Implement designed security functions and features into the system.
- Assess Security Effectiveness; Assess the effectiveness of ISSE activities.

Enterprise Security Architecture (ESA):
- Presents a long-term, strategic view of the system
- Unifies security controls
- Leverages existing technology investments

Implement Fail-Safe Design:
To implement fail-safe design, make sure that your software:
- Denies access by default in error-handling logic for security controls. Failure should not result in elevated rights for an attacker.
- Put limits on recovery retry attempts. If your software continually attempts to do something that isn't working, it may overfill caches, bog the process down trying to retry overwhelming numbers of backed-up tasks, and so forth.
- Doesn't make assumptions about ways to remediate when failure occurs. Fail bad inputs rather than attempting to correct them when you have no way to know what was intended. Suspend the affected transaction and report it, so users and system operators are clear that the transaction did not go through.

Integrate Security Into Your Development Processes:
- Examine each phase or aspect of your development processes, and identify how you can address security.
- Make sure security is included in your business requirements, software requirements specifications, and any other documentation and tools you use to define the scope and requirements of the project.
- Identify functional as well as non-functional security requirements, and make sure that these security requirements flow into test cases.
- Use a threat modeling process ("architectural risk analysis") to identify specific risks and prioritize how you will handle them.
- Include security reviews in your development phase and use code analysis tools to help identify security defects as they emerge.
- Use a variety of testing methods throughout development to ensure that security problems don't appear as the project progresses.
- As you fix security defects, create new automated unit tests to alert you if the problem you fixed reappears.
- Establish processes and software features so you are notified immediately when security issues are found.
- Establish ongoing monitoring and testing to identify when new security issues emerge over time.
- Stay apprised of vulnerabilities in systems and modules that your code depends on, such as the operating system it runs on, web servers, database servers, cloud services, and so forth.

Secure the Development Environment:
• Keep development, testing, and production environments completely separated, and control access to them by network users and deployment scripts through different assignable roles.
• If possible, separate responsibilities and access to different environments so only those who require access to a particular environment have access.
• Protect development, testing, and production environments from physical access.
• Protect the computers that developers use for development. For example, encrypt local storage, require developers to log out when walking away from their desk, and so forth.
• Avoid using public code repositories.
• Make sure your code repository is on a secure system that is protected from unauthorized physical and network access.
• Store code backups only in a secure storage location.
• Maintain secure logs of all code check-ins and check-outs.
• Monitor who accesses the repository, when, and from where.
• Audit code frequently to verify that no malicious functions or vulnerabilities are added into production.
• Provide developers with realistic sample data to use for programming and testing instead of actual data.
• When it is necessary to destroy any source code, sensitive data, assets, or backups, make sure they are destroyed securely. Follow compliance requirements, if applicable.

Perform Code Analysis:
When you perform static or dynamic code analysis:
• Combine static and dynamic code analysis to reveal more security defects than performing either type of code analysis alone.
• Combine automated code analysis with static and dynamic code analysis to reveal even more security defects.

Perform Static Analysis:
• Recognize benefits of, and uses for, static code analysis:
• Quick operation, functioning much faster than a manual (human) code reviewer.
• Scalable, can be run frequently (at each daily build, for example).
• Robotic consistency and rigor in checking for specific types of security problems.
• Low cost to operate, typically at a much lower cost than using experienced security architects and reviewers (whose efforts can be reserved for analysis tasks that benefit from human insight and creativity).
• Ability to quickly scan for a huge range of problems, drawing the developer's focus to potential problem areas.
• Recognize limitations of static analysis:
• May produce false negatives (not reporting problems that actually exist) and false positives (reporting problems that don't actually exist).
• Inability to identify certain kinds of security problems, such as authentication and access control problems and incorrect use of cryptography APIs.
• Inability to identify some problems due to other data values or resources not represented in code, such as misconfiguration of the host platform.
• Inability to analyze some code that would not be able to compile due to missing libraries, incomplete code, missing resources, and so forth.
• May provide a false sense that all security problems have been found.

Perform Dynamic Analysis:
• Recognize benefits of, and uses for, dynamic code analysis:
• Analyze code functioning in real world scenarios, minimizing the need to create artificial scenarios to find errors.
• Find certain types of vulnerabilities that static code analysis might not find, such as race conditions.
• Validate findings in the static code analysis.
• Recognize limitations of dynamic analysis:
• May produce false negatives (not reporting problems that actually exist) and false positives (reporting problems that don't actually exist).
• May provide a false sense that all security problems have been found.
• Require the code to run, so they can't identify issues in code that won't compile.
• Typically require more expertise than static code analysis to perform properly.
• Depend on scripts to automate tasks or users manually performing steps, so you can't guarantee full coverage of the source code.

Perform Automated Security Testing:
• Use automated testing to supplement, rather than replace, manual testing and code review.
• To support test-driven development (TDD), create security tests within your automated unit tests to ensure that security tests are continually performed during the development process.
• As much as possible, try to design tests to be repeatable across projects to save time, provide consistency, and facilitate testing process improvements over time.

The Systems Development Life Cycle:

• Initiation (considers value, sensitivity, regulatory compliance, classification, etc. of application/data).
• Define Functional Requirements (documents user and security needs).
• Design Specifications (system architecture/software designed).
• Development/Implementation/Testing (source code and test cases generated, quality/reliability addressed).
• Documentation/Program Controls (controls related to editing data, logging, version, control, integrity checks, etc.).
• Certification/Accreditation (independently testing data/code ensuring requirement are met, data validation, bounds checking, sanitizing, management's authorization for implementation).
• Production/Implementation (systems are live).

Build Security into Your Design Processes:

• Be sure that you understand what you are trying to build. Some developers document the software concept in a "theory of operations" document that describes what the software will do, and how it will do it. This may be recorded in more detail in requirements documents.
• Identify the environment in which your software will run.
• Identify the major modules in your software.
• List all of the errors that might occur in various modules, and how you will deal with them.
• Resist adding features that are not driven by requirements.
• Obtain, read, and follow secure coding standards defined for the specific programming languages and environments you use

SDLC:

• Project initiation and planning
• Functional requirements definition
• System design specifications
• Development and implementation
• Documentation and common program controls
• Testing and evaluation control, (certification and accreditation)
• Transition to production (implementation)

SDLC:

• Request/Gather information
  •• Security risk assessment
  •• Privacy risk assessment
  •• Risk-level acceptance
  •• Informational, functional, and behavioral requirements
• Design
  •• Attack surface analysis + Threat modeling
• Develop
  •• Automated CASE tools + Static analysis
• Test/Validation
  •• Dynamic analysis + Fuzzing + Manual Testing
  •• Unit, integration, acceptance, and regression testing
• Release/Maintenance
  •• Final security review

Note: Fuzz testing used to describe the use of known bad or randomized inputs to determine what unintended results may occur.

SDLC:

• Initiation- Identifying the need for a project
• System Concept Development- Defining the project scope and boundaries
• Planning- Creating the project management plan
• Requirements Analysis- Defining user requirements
• Design- Creating a Systems Design Document that describes how to deliver the project
• Development- Converting the design into a functional system
• Integration and Test- Verifying that the system meets the requirements
• Implementation- Deploying the system into the production environment
• Operations and Maintenance- Monitoring and managing the system in production
• Disposition - Migrating the data to a new system and shutting the system down

Note: The system life cycle (SLC) extends beyond the SDLC to include two:

• Operations and maintenance support (post-installation).
• Revisions and system replacement.

Development Methodologies:

• Build and fix

Lacks architecture design. Problems are fixed as they occur. Lacks a formal feedback cycle. Reactive instead of proactive.

• Waterfall

Linear sequential lifecycle. Each phase is completed before continuing.

Lacks a formal way to make changes during a cycle. The project is completed before collecting feedback and starting again.

• V-shaped

Based on the waterfall model. Each phase is complete before continuing.

Allows for verification and validation after each phase. Does not contain a risk analysis phase.

• Prototyping

Rapid prototyping uses a quick sample to test the current project. Evolutionary prototyping uses incremental improvements to design. Operational prototypes provide incremental improvements but are intended to be used in production.

• Incremental

Uses multiple cycles for development like multiple waterfalls. The entire process can restart at any time as a different phase. Easy to introduce new requirements. Delivers incremental updates to the software.

• Spiral

Continual approach to development. Performs risk analysis during development.

Future information and requirements are guided into the risk analysis. Allows for testing early in development.

• Rapid Application Development

Uses rapid prototyping. Designed for quick development. Analysis and design are quickly demonstrated. Testing and requirements are often revisited.

• Agile

Umbrella term for multiple methods. Highlights efficiency and iterative development.

User status describes what a user does and why. Prototypes are filtered down to individual features.

Systems Development Life Cycle:

• Initiation: During the initiation phase, the need for a system is expressed and the purpose of the system is documented.
• Development/Acquisition: During this phase, the system is designed, purchased, programmed, developed, or otherwise constructed.
• Implementation/Assessment: After system acceptance testing, the system is installed or fielded.
• Operation/Maintenance: During this phase, the system performs its work. The system is almost always modified by the addition of hardware and software and by numerous other events.
• Disposal: Activities conducted during this phase ensure the orderly termination of the system, safeguarding vital system information, and migrating data processed by the system to a new system, or preserving it in accordance with applicable records management regulations and policies.

Systems Development Life Cycle:

• Conceptual definition
• Functional requirements determination
• Control specifications development
• Design review
• Code review walk-through
• System test review
• Maintenance and change management

Insecure Code Practices:

• Comments in source code
• Lack of error handling
• Overly verbose error handling
• Hard-coded credentials
• Race conditions
• Unauthorized use of functions/unprotected APIs
• Hidden elements
• Sensitive information in the DOM
• Lack of code signing

Dynamic Code Analysis:

• Finds problems in code while the code is executing.
• Like static analysis, can be very helpful to see the source of quality and security defects.
• May be performed manually as a series of testing steps by a developer or tester working in the software development environment.
• Debuggers are a good tool for analyzing code as it runs.
• The dynamic analysis may also be scripted and monitored using automated testing tools.

Security Considerations in SDLC:

• Prepare a Security Plan
• Initiation
  •• Survey & understand the policies, standards, and guidelines
  •• Identify information assets (tangible & intangible)
  •• Define information classification & the protection level
  •• Define rules of behavior & security
  •• Conduct preliminary risk assessment
• Development/Acquisition
  •• Determine Security Requirements
  •• Conduct risk assessment
  •• Perform cost/benefit analysis
  •• Incorporate Security Requirements into Specifications
  •• Security planning (based on risks & CBA)
  •• Obtain the System and Related Security Activities
  •• Develop security test
• Implementation
  •• Install/Turn on Controls
  •• Security Testing
  •• Perform Security Certification & Accreditation of target system.
• Operation/Maintenance
  •• Security Operations and Administration
  •• Operational Assurance
  •• Audits and Continuous monitoring
  •• Configuration management & performs change control
• Disposal
  •• Information transfer or destruction
  •• Media Sanitization
  •• Dispose of hardware

Identify Sources of Security Requirements:

• User expectations
• Standards and compliance requirements
• Business requirements
• Requirements for platforms, services, and APIs that your software uses
• Identification of where your software is vulnerable, and identify how you will address each vulnerability

Positive/Negative Test:

• Positive Test - Work as expected (Output as per given input - goes as per plan)
• Negative Test - Even unexpected inputs are handled gracefully with tools like Exception Handlers

Coverage Testing:

For analyzing, you should be aware of the following coverage testing types:
• Black box testing: The tester has no prior knowledge of the environment being tested.
• White box Testing: The tester has full knowledge before testing.
• Dynamic Testing: The system that is being tested is monitored during the test.
• Static Testing: The system that is being tested is not monitored during the test.
• Manual Testing: Testing is performed manually by hands.
• Automated Testing: A script performs a set of actions.
• Structural Testing: This can include statement, decision, condition, loop, and data flow coverage.
• Functional Testing: This includes normal and anti-normal tests of the reaction of a system or software. Anti-normal testing goes through unexpected inputs and methods to validate functionality, stability, and robustness.
• Negative Testing: This test purposely uses the system or software with invalid or harmful data, and verifies that the system responds appropriately

Code Repository Security:
• System security
• Operational security
• Software security
• Secure communications
• File system and backups
• Employee access
• Maintaining security
• Credit card safety

The Life Cycle of any Process:
• Plan and organize
• Implement
• Operate and maintain
• Monitor and evaluate

Regression and Acceptance Testing include:
• Test fixed bugs promptly.
• Watch for side effects of fixes.
• Write a regression test for each bug fixed.
• If two or more tests are similar, determine which is less effective and get rid of it.
• Identify tests that the program consistently passes and archive them.
• Focus on functional issues, not those related to design.
• Make changes (small and large) to data and find any resulting corruption.
• Trace the effects of the changes on program memory.

RUM vs. Synthetic:
• RUM harvests information from actual user activity, making it the most realistic depiction of user behavior.
• Synthetic monitoring approximates user activity, but is not as exact as RUM

Software Acquisition:
• Planning
• Contracting
• Monitoring
• Acceptance
• Follow on

Software Requirements:
• Informational model
• Functional model
• Behavioral model

Software Protection Mechanisms:
• Security Kernels
• Processor privilege states
• Security controls for buffer overflow
• Controls for incomplete parameter check and enforcement
• Memory protection
• Covert channel controls
• Cryptography
• Password protection techniques

API formats:
• Representational State Transfer (REST) - is a software architecture style, consisting of guidelines and best practices for creating scalable web services.
• Simple Object Access Protocol (SOAP) - is a protocol specification for exchanging structured information in the implementation of web services in computer networks.

API Security:
• Use same security controls for APIs as for any web application on the enterprise.
• Use Hash-based Message Authentication Code (HMAC).
• Use encryption when passing static keys.
• Use a framework or an existing library to implement security solutions for APIs.
• Implement password encryption instead of a single key-based authentication.

Forensic:
The forensic investigation process must demonstrate that information handling procedures and actions performed did not alter the original data throughout the custody chain. This may include:
• Recording the name and contact information of those charged with maintaining a chain of custody
• Details of the timing of the event
• Purpose of moving the data
• Identification of evidence through recording of serial numbers and other details
• Sealing the evidence with evidence tape
• Documenting the location of storage
• Documenting the movement of the information

Concepts unique to the forensic analysis:
• Authorization to collect information
• Legal defensibility
• Confidentiality
• Evidence preservation and evidence security
• Law enforcement involvement

Forensic Process:
• Identification
• Preservation
• Collection
• Examination
• Analysis
• Presentation
• Decision

Generic Computer Forensic Investigation Model:
• Pre-process
• Acquisition and preservation
• Analysis
• Presentation
• Post-process

E-discovery Process:
• Information Governance
• Identification
• Preservation
• Collection
• Processing
• Review
• Analysis
• Production
• Presentation

CSIRT:
Organizations will often form a cybersecurity incident response team (CSIRT) to help identify and manage information security incidents. The individuals that make up the CSIRT are trained in proper collection and preservation techniques for investigating security incidents. National Institute of Standards and Technology Special Publication (NIST SP) 800-61r2 identifies the following models for organizing such a team.
• Central team One team handles incidents on behalf of the entire organization.
• Distributed team For larger or geographically dispersed organizations, it may be more appropriate to have individual CSIRTs for different segments of the organization or different geographic locations.
• Coordinating team An overarching central team can be added to provide guidance and coordination among distributed teams.

CSIRT Tools:
The CSIRT has a number of tools they can use to help handle security incidents. Keeping the toolkit up-to-date will contribute to the CSIRT working optimally. The following table lists a few common examples.
• The Sleuth Kit (TSK) / Cross-platform
• EnCase / Windows
• Forensic Toolkit (FTK) / Windows
• Forensics Explorer / Windows
• SANS Investigative Forensic Toolkit (SIFT) / Ubuntu (Linux)
• Digital Forensics Framework (DFF) / Cross-platform
• Computer Online Forensic Evidence Extractor (COFEE) / Windows
• WindowsSCOPE / Windows
• HashMyFiles / Windows
• Volatility / Windows, Linux
• TestDisk / Cross-platform
• Wireshark / Cross-platform

Data Classification Scheme:
• Identify custodian
• Specify evaluation criteria
• Classify and label each resource
• Document any exceptions
• Select security controls
• Specify the procedures for declassifying
• Create enterprise awareness program

Data Classification:
• Scope (value, Age)
• Classification Controls
• Assurance
• Marking and labeling

Classify Information:
• Specify the classification criteria
• Classify the data
• Specify the controls
• Publicize awareness of the classification controls

Classification program:
• Define classification level
• Identify owner
• Determine security level
• Develop a procedure to declassifying

Data Classification Procedures:
• Define classification levels.
• Specify the criteria that will determine how data are classified.
• Identify data owners who will be responsible for classifying data.
• Identify data custodian who will be responsible maintaining data and sec. level.
• Indicate the security controls, protection mechanisms, required for each class level
• Document any exceptions to the previous classification issues.
• Indicate the methods that can be used to transfer custody of info to diff owner.
• Create a procedure to periodically review the classification and ownership.
• Communicate any changes to the data custodian.
• Indicate procedures for declassifying the data.
• Integrate these issues into the security-awareness program.

Data Collection Limitations:
• Data collection only for legal and fair means.
• Data collection with the knowledge and approval of the subject.
• Do not use personal data for other purposes.
• Collection of personal data should be relevant for the purpose.
• Collected data to be accurate and kept up to date.
• Do not disclose personal data with other parties without the permission of the subject.
• Secure personal data against intentional or unintentional access, use, disclosure,
destruction, and modification.
Note: The following are some of the important privacy-related practices and rules across the
world that provide frameworks and limitations relating to personal data.
• General Data Protection Regulation (European Union)
• Data Protection Directive (EU)
• Data Protection Act 1998 (U.K)
• Data Protection Act, 2012 (Ghana)
• Data protection (privacy) laws in Russia
• Personal Data Protection Act 2012 (Singapore)
• Privacy Act (Canada)

The goal of Incident Handling and Response Planning:
• Detects compromises as quickly and efficiently as possible.
• Responds to incidents as quickly as possible.
• Identifies the cause as effectively as possible.

Purpose of Incident Response:
• Restore normal service
• Minimize impact on business
• Ensure service quality and availability are maintained

Incident Response:
• Triage (assesses the severity of the incident and verify)
• Investigation (contact law enforcement)
• Containment (limit the damage)
• Analysis
• Tracking

Incident Response:
• Preparation
• Detection -- Identification
• Response -- Containment
• Mitigation
• Reporting -- Report to Sr. Management
• Recovery -- Change Management & Configuration. Management
• Remediation -- RCA & Patch M. & Implement controls
• Lessons Learned -- Document and knowledge transfer

Incident Response:
- Preparation
- Detection
- Containment
- Eradication
- Recovery
- Post Incident Review/Lesson learned

Incident Handling Steps (NIST 800-61):
- Preparation People
- Identification Identify
- Containment Containers
- Eradication Ending
- Recovery Real
- Lessons Learned Lives

Incident Response Process (PIC-ERL):
- Preparation
- Identification
  - Detection/analysis
  - Collection
- Containment
- Eradication
- Recovery
- Post-incident
  - Lessons learned
    - ••••• Root cause analysis
  - Reporting and documentation

Note: Gap analysis includes reviewing the organization's current position/performance as revealed by an audit against a given standard.

Incident Response Process:
- Plan for and identify the incident.
- Initiate incident handling protocols.
- Record the incident.
- Evaluate and analyze the incident.
- Contain the effects of the incident.
- Mitigate and eradicate the negative effects of the incident.
- Escalate issues to the proper team member, if applicable.
- Recover from the incident.
- Review and report the details of the incident.
- Draft a lessons-learned report.

Incident Response Plans:
A usable IR plan is dynamic enough to address many incidents, but simple enough to be useful. Some characteristics of a plan are:
- Brief During an incident, there is little time to read and understand large documents and find highlighted portions that may be relevant.
- Clear Incidents are complex and often, are not well understood in the beginning.
- Resilient Rigid and prescriptive incident response plans can fail when key participants are absent.
- Living This is not just a plan to be reviewed and (potentially) updated once annually.

Incident Response Plans Models:
Compliance Driven:
- Designed to evaluate a response after the fact.
- Reflects an approach from an audit and compliance (HIPAA, GLBA, PCI-DSS).
- Security engineers and analysts do not refer to them during an incident, except possibly in retrospective reports.

Technical Driven:
- Elaborate playbooks that communicate techniques for data analysis and are often unwieldy and intentionally vague about accountability.
- Developed by security or network engineers, but can be frustrating when evaluating a response to reports to the Board of Directors or executives.

Coordinated (Compliance Driven + Technically Driven):
- Provides a framework for activities where they are more ambiguous: between teams and roles. The coordinated plan describes communication and authority so they are not in question during an incident, but also allows the expertise of a team to be applied without micromanagement by the plan.

Incident Investigation Methodology:
- Analysis and Imaging
- Dead box forensics
- Volatile data collection
- Server handling
- Endpoint imaging
- Live system handling (Volatile data collection)
- Write-block
- Controlled forensic boot (Volatile data considerations)

Respond Appropriately to Data Breaches:
A data breach should be followed up with an appropriate response. For example, you should limit the extent of the leak, you should inform those who are affected, and you should remedy any defects or problems that made the breach possible. To avoid this defect:
- Provide continuous monitoring and logging features to monitor for situations that might indicate personal data leakage and loss.
- Provide features to warn users of possible suspicious activity in their accounts.
- Create, maintain, and periodically test an incident response plan.
- Continuously monitor for personal data leakage and loss.
- When a breach occurs:
- Validate that the breach occurred.
- Determine the most effective way to prevent further leakage, and implement it.
- Assign an incident manager to be responsible for the investigation.
- Decide how to investigate and respond to the data breach to ensure that evidence is appropriately handled.
- Assemble an incident response team.
- Notify affected people as appropriate.
- Determine whether to notify the authorities as appropriate.
- Remedy any defects or problems that made the breach possible.

Visibility challenges:
• Discovering and monitoring assets
• Seeing and protecting end-user devices off the network
• Finding vulnerabilities in application code that the organization builds itself
• Identifying weaknesses in IoT devices that could lead to compromise
• Assessing critical infrastructure systems without disrupting operations

Information Security Continuous Monitoring:
• Define
• Establish
• Implement
• Analyze
• Respond
• Review
• Update
• Repeat

Capture Security Requirement:
• Threat modeling
• Data classification
• Risk assessments

Threat modeling:
Works to identify, communicate, and understand threats and mitigations within the context of protecting assets of value. STRIDE threat model: System for classifying known threats based on the kinds of exploits used or the motivation of the attacker.

Threat Modelling:
• Assessment scope
• System Modeling
• Identify Threat
• Identify Vulnerability
• Exam Threat history
• Impact
• Response

Threat modeling: (STRIDE):
• Spoofing: Attacker assumes the identity of the subject
• Tampering: Data or messages are altered by an attacker
• Repudiation: Illegitimate denial of an event
• Information Disclosure: Information is obtained without authorization
• Denial of Service: Attacker overload system to deny legitimate access
• Elevation of Privilege: Attacker gains a privilege level above what is permitted
Threat

DREAD:
The Microsoft DREAD ranking model builds upon the traditional risk model: Risk = Likelihood x
Impact. For example, suppose you evaluated a particular threat and assigned a 10-point value to
each of the following questions as shown.
• Ease of Exploitation:
• Discoverability—How easily can an attacker discover this threat? (8, relatively easy)
• Reproducibility—How easy is it to reproduce an attack to work? (10, very easy)
• Exploitability—How much time, effort, and expertise is needed to exploit the threat? (7, relatively easy)
• Impact:
• Affected Users—What percentage of users would be affected? (10, affects all users)
• Damage—How great would the damage be in a successful attack? (9, very high)

Threat Modeling:
• Assessment Scope
• System Modeling
• Identify Threats
• Identify Vulnerabilities
• Examining the Threat History
• Evaluation of Impact on the Business
• Developing a Security Threat Response Plan

Threat Modeling Tools:
• Microsoft - Threat Modeling Tool
• MyAppSecurity - Threat Modeler
• IriusRisk Threat - Modeling Tool
• Scandinavian - securiCAD
• Security Compass - SD Elements

Threat Modeling Process:

- Define general security objectives and scope
  - Know your assets/data (not just physical).
  - Collect data such as existing documentation, policy, framework, guideline, DB, users stories, errors Check the accuracy of the collected data.
  - Gather security requirements already defined for you via compliance, government regulations, and industry standards.
  - Identify how you can address security and implement security requirements from a regulatory/data privacy perspective.
- Decompose
  - Know your organization connectivity models. Ensure that no elements have been forgotten by identifying sub-components, dependencies and interaction points.
  - Identify assets an attacker might be interested in, who should be allowed to access each area, and how access is controlled.
  - Break up your application/system into conceptual entry points, components, and boundaries where an attacker might interact with it.
  - Mark all untrusted data inputs.
  - Diagram how data flows through the application/system using data flow diagrams (DFDs). DFD will represent how data moves between processes, storage, and external systems/services.
- Identify and rank threats
  - List all threats categories as possible, e.g., reconnaissance, social engineering, systems hacking, web-based threats, malware, hijacking and impersonation, denial of service Mobile-based threats, cloud-based threats, etc.
  - Existing threats should be identified before controls are listed for each threat, but the ranking of those threats will determine which controls will be implemented. Ranking threats is a key because the likelihood or impact of a threat may be so low that performing a control is not worth the cost.
  - Think like an attacker. It is terrible to crash, but it is worse to have wrong information and not even know about it. Examine your application and identify where threats exist such as checking return codes, errors, level of access, data sharing and all input if possible.
  - Ensure security requirements flow into test cases.
  - Use root cause analysis.
  - Use methodology like STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Escalation of Privileges) to help you identify and rank threats.
  - If you are using third-party components, libraries and services consider and include their own threat models.
- Counter each threat
  - Follow security design patterns to deal with specific types of threats.
  - Provide countermeasures for each threat you need to address.
  - Establish ongoing monitoring to identify when new security issues arise over time.
  - Test the mitigation, in case threats not mitigated become security bugs in your bug repository.

Cyber Kill Chain:

- Reconnaissance
- Weaponization
- Delivery
- Exploitation
- Installation
- Command and Control
- Actions on Objectives

The Cyber Security Operational Life Cycle:

- DISCOVER: Identify and map every asset across any environment. From here you can baseline the current and desired operational state.
- ASSESS: With every change, automatically assess the current state against the baseline state of the environment, including misconfigurations, vulnerabilities and other key indicators of security health, such as out of date antivirus or high risk users.
- ANALYZE: Add context to the asset's exposure to prioritize remediation based on the asset's business criticality and the severity of the vulnerability.
- FIX: Prioritize which exposures to fix first, if at all, and select the appropriate remediation technique, whether it's a temporary security control or a complete fix.

Data Exfiltration:

- Covert channels
- File sharing services

TOCTOU:

A type of race condition called Time of Check to Time of Use because the problem arises when shared data is changed between the time when it is initially checked, and when it is used. Race conditions are often non-deterministic, meaning that you can't predict the outcome since it is based on timing. Race conditions are often hard to debug, since running in a debugger adds timing delays that change the outcome. Prevent race conditions by preventing multiple simultaneous requests (locking) or through a synchronization mechanism.

Storage vs. Timing Channels:

Covert channels can also be thought of in terms of two different categories: storage and timing. A covert storage channel includes one process writing to a storage location and another process reading from that location. A covert timing channel includes one process altering system resource so that changes in response time can signal information to the recipient process. Some usage of covert channels combines both aspects of storage and timing.

Examples of covert channels include the following:
• Transmitting data over a rarely used port that the firewall does not block.
• Concealing data in the headers of TCP/IP packets so as to evade signature analysis by intrusion
detection systems.
• Breaking the data up into multiple packets to be sent at different times in order to evade
signature analysis.
• Transmitting data over a shared resource that is not typically used as a communication channel
(i.e., file system metadata).
• Transmitting encrypted data that cannot be inspected as it leaves the network.

Steganography:
Similar to using a covert channel, one technique for hiding data for exfiltration is steganography. Using steganography, an attacker might be able to evade intrusion detection and data loss countermeasures if they hide information within images or video. Modern tools hide digital information so well that the human eye cannot tell the difference; likewise, computer programs not equipped for steganographic analysis may also fail to spot the hidden information.

Information Systems Auditor:
• Audits information security activities for compliance; Verifies adherence to security objectives, policies, procedures, standards, regulations, and related requirements.
• Verifies whether information security activities are managed and operated to ensure achievements of state security objectives.
• Provides independent feedback to senior management.

Auditing uses:
• Record review
• Adequacy of controls
• Compliance with policy
• Detect malicious activity
• Evidence of persecution
• Problem reporting and analysis

Audit:
The systematic process by which a competent, independent person objectively obtains and evaluates the evidence regarding assertions about an economic entity or event for the purpose of forming an opinion about and reporting on the degree to which the assertion conforms to an identified set of standards. Audit: Evaluate security controls - Report on their effectiveness - Recommend improvements

Audit Plan:
• Define audit objectives
• Define the audit scope
• Conduct audit
• Refine the audit process

Audit Process:
• Determine goals
• Involve right business unit leader
• Determine Scope
• Choose audit Team
• Plan audits
• Conduct audit
• Document result
• Communicate result

Audit Report:
• Purpose
• Scope
• Results discovered or revealed by the audit
• Problems, events, and conditions
• Standards, criteria, and baselines
• Causes, reasons, impact, and effect
• Recommended solutions and safeguards

IT security audit is designed to find:
• Malfunctioning controls
• Inadequate controls
• Failure to meet target standards/guidelines

Software-Defined Everything (SDx):
Extension of virtualization that abstracts an application or function from its underlying hardware, separating the control and data planes and adding programmability. Beginning with software-defined networking (SDN), SDx now encompasses software defined storage (SDS), software-defined computing, software-defined security, and software-defined data centers (SDDC), among others.

Software-Defined networking (SDN):
• Application
• Control
• Infrastructure

Software-Defined networking (SDN):
• Network administrators can adjust network traffic on the fly.
• Provide the ability to better detect network traffic anomalies.
• They add a higher level of complexity to the network that requires special skills.

Communication Characteristics Asynchronous:
• No timing component
• Surrounds each byte with processing bits
• Parity bit used for error control
• Each byte requires three bits of instruction (start, stop, parity)

Communication Characteristics Synchronous:
• Timing component for data transmission synchronization
• Robust error checking, commonly through cyclic redundancy checking (CRC)
• Used for high-speed, high-volume transmissions
• Minimal overhead compared to asynchronous communication

Networking Hardware:
• Modems (converts digital to analog/analog to digital signals)
• Hubs (operate at the physical layer, retransmit signals)
• Repeaters (operate at the physical layer, re-amplify signals)
• Bridges (operate at layer 2, filters traffic)
• Switches (operate at layer 2, forwards broadcasts and frames)
• Routers (forwards packets)

Content-Distribution Network (CDN) benefits:
• On-demand scaling
• Cost efficiency
• Locality of Content
• Security Enhancement
• Filter out DDOS attacks

The main protocols of IPSec suite:
• Authentication Header (AH) Provides data integrity, data origin authentication, and protection from replay attacks
• Encapsulating Security Payload (ESP) Provides confidentiality, data origin authentication, and data integrity
• Internet Security Association and Key Management Protocol (ISAKMP) Provides a framework for security association creation and key exchange
• Internet Key Exchange (IKE) Provides authenticated keying material for use with ISAKMP

Point-to-Point Tunneling Protocol (PPTP):
• Works in a client/server model
• Extends and protects PPP connections
• Works at the data link layer
• Transmits over IP networks only

Layer 2 Tunneling Protocol (L2TP):
• Hybrid of L2F and PPTP
• Extends and protects PPP connections
• Works at the data link layer
• Transmits over multiple types of networks, not just IP
• Combined with IPSec for security

IPSec:
• Handles multiple VPN connections at the same time
• Provides secure authentication and encryption
• Supports only IP networks
• Focuses on LAN-to-LAN communication rather than user-to-user communication
• Works at the network layer, and provides security on top of IP

Transport Layer Security (TLS):
• Works at the session layer and protects mainly web and e-mail traffic
• Granular access control and configuration are available
• Easy deployment since TLS is already embedded into web browsers
• Can only protect a small number of protocol types

Drawbacks multilayer protocols:
• Covert channels are allowed
• Filters can be bypassed
• Logically imposed network segment boundaries can be overstepped

Benefits multilayer protocols:
• A wide range of protocols can be used
• Encryption
• Flexibility and resiliency

MPLS feature:
• Traffic engineering
• Better router performance
• Built-in tunneling

Two main MPLS routing protocols:
• Label Distribution Protocol (LDP) - No Traffic Engineering
• Resource Reservation Protocol with Traffic Engineering (RSVP-TE)

Label Switched Path (LSP) MPLS Router Roles/Positions are:
• Label Edge Router (LER) or "Ingress Node" - The router that first encapsulates a packet inside an MPLS LSP; Also the router that makes the initial path selection.
• Label Switching Router (LSR) or "Transit Node" - A router that only does MPLS switching in the middle of an LSP.
• Egress Node - The final router at the end of an LSP, which removes the label.

Generic Routing Encapsulation (GRE) Tunnel
Tunneling protocol developed by Cisco that can encapsulate a wide variety of network layer protocols inside virtual point-to-point links over an internet protocol network.

Data Center Site Infrastructure Tier Standard Topology:
Four-tiered architecture, each progressively more secure, reliable, and redundant:
- Tier 1: Basic data center site infrastructure (basic protection)
- Tier 2: Redundant site infrastructure capacity components
- Tier 3: Concurrently maintainable site infrastructure
- Tier 4: Fault-tolerant site infrastructure (life-dependent applications and services)

Temperature and Humidity Guidelines:
American Society of Heating, Refrigeration, and Air Conditioning Engineers (ASHRAE) Technical Committee 9.9 provides guidelines for data center temperature and humidity.
- Temperature: 64.4-80.6°F, 18-27°C (at equipment intake)
- Humidity: 40% @ 41.9°F (5.5'C) to 60% @ 59°F (15°C)

Common Criteria CC:
• PP - what the customer needs
• ST - what Vendor provides
• TOE - The actual product
• EAL - Rating which provides Evaluation and Assurance
Note: The EAL is a measure of how thoroughly the security features the product vendor claims the product offers have been tested and reviewed, and by whom. The EAL does not offer any true measure of how well those security features will work in a production environment, whether those features are preferable to other features offered by competing products, or whether the product is "good."

EAL:
• EAL1 - Functionally tested (lowest rating)
• EAL2 - Structurally tested
• EAL3 - Methodically tested and checked
• EAL4 - Methodically designed, tested and reviewed (medium rating)
• EAL5 - Semi-formally designed and tested
• EAL6 - Semi-formally verified, designed and tested
• EAL7 - Formally verified, designed and tested (highest rating)

Before selecting a Security Monitoring Tool type:
• It should collect information from numerous sources.
• It should be able to inter-operate with other systems, such as a help desk or change management program.
• It should comply with all relevant laws and industry regulations.
• It should offer scalable reporting so you get both a high-level and low-level perspective on your security

Security Information and Event Management (SIEM):
• Correlation
• Compliance
• Alert

What SIEMs Provide:

- Data aggregation: Bringing many logs from operating system, network devices, and applications together for analysis
- Correlation: Looking for common attributes within the logs that may be used to chain together events
- Alerting
- Dashboards: Much quicker than reading through reports
- Compliance: Can generate compliance reports based on event log data
- Retention: Long-term storage - Most SIEMs don't provide long-term storage in an active manner. They tend to offload events after a certain age to an internal archival area. This is due to the fact that you could end up with billions upon billions of events over time, and most systems cannot manage that much data efficiently.
- Forensic analysis: Searching through logs from many systems by specific date, time, or other criteria

Tasks may be performed automatically for you with tools such as SIEMs:

• Filter out unnecessary or duplicate data
• Combine sources
• Synchronize events logged in different sources
• Normalize data formats
• Store data securely
• Data Collection, Analysis, and Correlation

SIEM on Cloud ....the benefits are:

• No capital expenditure
• No need to invest on premise machines
• No need to invest in technical support for hardware
• No installation charges
• Only fine tuning
• Upgrades rolled out automatically by the cloud provider

Security Mode:

• Dedicated security mode (All users can access all data).
• System high-security mode (on a need-to-know basis, all users can access limited data).
• Compartmented security mode (on a need-to-know basis, all users can access limited data as per the formal access approval).
• Multilevel security mode (on a need-to-know basis, all users can access limited data as per formal access approval and clearance).

Prevent SQL Injection (SQLi):

• Perform Input Validation
• Limit Account Privileges
• Use Stored Procedures

In a SQL injection attack, an attacker could:

• Harvest and crack password hashes
• Delete and modify customer records
• Read and write system files

Injection attacks:

SQL injection attack consists of insertion or "injection" of a SQL query via the input
• HTML injection is a type of injection issue that occurs when a user is able to control an input point and is able to inject arbitrary HTML code into a vulnerable web page
• Command injection is an attack in which the goal is the execution of arbitrary commands on the host operating system via a vulnerable application
• Code injection allows the attacker to add his own code that is then executed by the application.

Web App Threats:

• Cookie Poisoning
• Insecure Storage
• Information Leakage
• Directory Traversal
• Parameter/Form Tampering
• DOS Attack
• Buffer Overflow
• Log tampering
• SQL Injection
• Cross-Site (XSS)
• Cross-Site Request Forgery
• Security Misconfiguration
• Broken Session Management
• DMZ attack
• Session Hijacking
• Network Access Attacks

DNSSEC:
Adds security to DNS by enabling DNS responses to be validated. DNSSEC uses a process called zone signing that uses digital certificates to sign DNS records.

Threats to DNS:

- Footprinting: An attacker attempts to gather all DNS records for a domain via domain transfer in order to map out the target environment.
- Denial of Service (DoS): Flooding of DNS servers can prevent the server from responding to DNS requests.
- Redirection: An attacker redirects queries to a server under the attacker's control.
- Spoofing: Also known as DNS poisoning where the attacker provides incorrect DNS information for a domain to a DNS server, which will then give out that incorrect information.

Social Engineering:
It's important for any user to understand social engineering and their tactics. Additionally, by understanding the underlying principles, it becomes easier to avoid being tricked by them. The following sections introduce these principles.

- Authority
- Intimidation
- Consensus / Social Proof
- Scarcity
- Urgency
- Familiarity/Liking
- Trust

Wireless and RF Vulnerabilities:

- Evil Twin
- Karma Attack
- Downgrade attack
- Dauth. Attack
- Fragmentation Attack
- Credential Harvesting
- WPS Implementation Weakness
- Bluejacking
- Bluesnarfing
- RFID Cloning
- Jamming
- Repeating

Basic MALWARE Analysis:

- Malware assessment
- String analysis
- Dependency analysis
- Encountering files with wiped logical data
- Sandbox analysis
- Online malware scanner / sandbox

Basic TCB function:

- Process activation
- Execution domain switching
- Memory protection
- I/O operation

Memory Manager:

- Relocation
- Protection
- Sharing
- Logically Organization
- Physical Organization

Memory Protection:

- DEP (Data Execution Prevention)
- ASLR (Address Space Layout Randomization)
- ACL (Access Control List)

Memory Protection:

- Segmentation
- Paging
- Protection keying

Attacks (Mitigation):

- Eavesdropping (encryption)
- Cyber-squatting (Secure your domain registration)
- SPAM (email filtering)
- Teardrop (patching)
- Overlapping fragment (not allowing fragments to overwrite)
- Source routing Attack (block source-routed packets)
- SYN flood Attack (vendor support in securing network stack)
- Spoofing (patching, firewalls, strong authentication mechanisms)
- Session hijacking (encryption, regular re-authentication)

Attacks Phase:
- Gaining Access
- Escalating Privileges
- System Browsing
- Install Additional Tools
- Additional Discovery

WLAN attacks:
- Confidentiality Attacks
  - Traffic Analysis
  - Eavesdropping
  - Man-in-the-Middle Attack
  - Evil Twin AP
- Access Control Attacks
  - War Driving
  - Rogue Access Point
  - MAC addresses spoofing
  - Unauthorized Access
- Integrity Attacks
  - Session Hijacking
  - Replay Attack
  - Frame Injection Attack
- Availability Attacks
  - Denial-of-Service Attack
  - Radiofrequency (RF) Jamming
  - Beacon Flood
  - Associate/Authentication Flood
  - De-authentication & Disassociation
  - Queensland DoS / Virtual carrier-sense attack
  - Fake SSID
  - AP theft
- Authentication Attack
  - Dictionary & Brute force attack

Securing WLANs:
- Change the default SSID.
- Implement WPA2 and 802.1X to provide centralized user authentication
- Use separate VLANs
- Deploy a wireless intrusion detection system (WIDS).
- Physically put the AP at the center of the building.
- Logically put the AP in a DMZ with a firewall between the DMZ and internal network.
- Implement VPN for wireless devices to use. This adds another layer of protection for data being transmitted.
- Configure the AP to allow only known MAC addresses into the network.
- Carry out penetration tests on the WLAN.

Threats to the DNS Infrastructure:
- Footprinting
- Denial-of-Service Attack
- Data modification
- Redirection
- Spoofing

Attacks against DNS servers:
- Zone transfer: Information gathering shortcut
- Zone poisoning: Breach primary server and alter the zone file to the corrupt domain
- Cache poisoning: Send false answers to cache servers until they store them
- Reflection DoS: Send bogus requests into a chain of servers that do recursive queries

Reduce XSS:
- Data validation
- Data Sanitization
- Cookies security
- Output Escaping

Facility Attacks
- Piggybacking
- Fence jumping
- Dumpster diving
- Lockpicking
- Lock bypass
- Egress sensor
- Badge cloning

Man-in-the-middle:
- ARP spoofing
- ICMP redirect
- DHCP spoofing
- NBNS spoofing
- Session hijacking
- DNS poisoning

Isolating CPU processes:
- Encapsulation of objects
- Time multiplexing of shared resources
- Naming distinctions
- Virtual memory mapping

Security mechanisms:
- I/O operations
- Process activation
- Domain switching
- Memory protection
- Hardware management

Hacking Website: (Deface Websites)
- SQL injection
- XSS / CSRF
- Remote file inclusion
- Local file inclusion
- DDOS
- Exploiting vulnerability
- Directory traversal
- Command injection

Emergency-Response Guidelines include:
- Immediate response procedures
- List of the individuals who should be notified of the incident
- Secondary response procedures that first responders should take

ISC2 - Code of Ethics:
- Protect Society, Commonwealth Infrastructure
- Act honorably, honestly, justly, responsibly and legally
- Provide diligent, competent service to the Principles
- Advance and protect the profession

Background Checks:
- Credit History
- Criminal History
- Driving Records
- Drug and Substance Testing
- Prior Employment
- Education, Licensing, and Certification Verification
- Social Security Number Verification and Validation
- Suspected Terrorist Watch List

Vulnerability management:
• Inventory
• Threat
• Asses
• Prioritize
• Bypass
• Deploy
• Verify
• Monitor

Vulnerability Assessment:
• Collect
• Store
• Organize
• Analysis
• Report

Consideration of vulnerability scanning:
• Time to run a scan
• Protocols used
• Network topology
• Bandwidth limitations
• Query throttling
• Fragile systems/non-traditional assets

Vulnerability Assessment and Pen Testing:
• Scope
• Information gathering
• Vulnerability detection
• Information analysis and planning
• Penetration testing
• Privilege escalation
• Result analysis
• Reporting
• Cleanup

Note: Vulnerability assessments should be done on a regular basis to identify new vulnerabilities. VA scanners usually don't have more than a Reading privilege.

Penetration Test:
• Discovery - Obtain the footprint and information about the target.
• Enumeration - Perform ports scans and resource identification.
• Vulnerability mapping - Identify vulnerabilities in systems and resources.
• Exploitation - Attempt to gain unauthorized access by exploiting the vulnerabilities.
• Report - Report the results to management with suggested countermeasures

Main sections defined by the standard as the basis for penetration testing execution:
• Pre-engagement Interactions
• Intelligence Gathering
• Threat Modeling
• Vulnerability Analysis
• Exploitation
• Post Exploitation
• Reporting

Penetration Test:
• Goal
• Recognizance
• Discovery
• Exploitation
• Brute-Force
• Social Engineering
• Taking Control
• Pivoting
• Evidence
• Reporting
• Remediation

Penetration Testing:
• External testing
• Internal testing
• Blind testing - Limited information on the PT team
• Double-blind testing - No information to the internal security team
• Targeted testing - Both internal and PT team aware.

Penetration Testing:
• Reconnaissance
• Scanning
• Gaining Access
• Maintaining Access
• Covering Tracks

Penetration Testing:
• Performing basic reconnaissance to determine system function
• Network discovery scans to identify open ports
• Network vulnerability scans to identify unpatched vulnerabilities
• Web application vulnerability scans to identify web application flaws
• Use of exploit tools to automatically attempt to defeat the system security
• Manual probing and attack attempts

Penetration Testing Key Components:
• Threat Emulation
• Attack Surface
• Attack Vectors
• Attack Scenarios
• Methodology

Penetration Testing Techniques:
• Wardriving/dialing
• Eavesdropping
• Network sniffing
• Physical security testing
• Social engineering

Penetration Testing Rules of Engagement:
• Identifies and fines the appropriate testing method(s) and techniques with exploitation of the relevant devices and/or services
• While scope defines the start and the end of an engagement, the rules of engagement define everything in between

Rules of engagement (ROE) in Pen Test:
• Introduction
• Logistics
• Communication
• Targets
• Execution
• Reporting
• Signatures

Types of Penetration Tests:
• Network Penetration Test
• Application Penetration Test
• Appliance / Internet Of Things (IoT) Penetration Test
• Enterprise Penetration Test
• Red Team
• Reverse Engineering / Zero-day Research

Penetration Testing:
• Requires one or more objectives for a successful test
• The scope is based on the attack scenarios
• The effort is 'time-boxed.'
• Discovers both technical and logical vulnerabilities
• Reports should be concise
• Recommendations are strategic
• Enhances internal security operations processes

There are a few elements that are common to most effective Pen Testing reports:
• Preparation:
•• Identify the objectives and purpose of the penetration test.
•• Consider how best to address the audience you are writing to.
•• Ensure that you can place all relevant events in the context of time.
• Content:
•• Detail the test methodology you used in your tests.
•• Detail the results of each test, identifying specific assets and vulnerabilities that you id
• Provide your analysis and interpretation of the results.
• Suggest remediation techniques to employ.
• Formatting:
•• Format your report to comply with all of the applicable gov. regulations and with standards.
•• Write in clear, practical language. Avoid technical jargon.
•• Format your report with groups and sections to enhance readability.
• Reviewing:
•• Proofread your document before sending it out.
•• Ask another expert to provide a second opinion on the report before sending it out.

Enumeration:
• Extracting usernames using emails IDs, default passwords
• Extracting usernames using SNMP
• Extracting information using DNS zone transfer, Finger OS, and ports

Scanning Types:
• DISCOVERY SCANNING: A discovery scan can be performed with very simple methods, for example, by sending a ping packet (ping scanning) to every address in a subnet. More sophisticated methods will also discover the operating system and the services of a responding device.
• COMPLIANCE SCANNING: A compliance scan can be performed either from the network or on the device (for instance, as a security health check). If performed on the network, it will usually include testing for open ports and services on the device.
• VULNERABILITY SCANNING: A vulnerability scan can either test for vulnerability conditions or try an active exploitation of the vulnerability. A vulnerability scan can be performed in a non-disruptive manner, or under acceptance of the fact that even a test for certain vulnerabilities might affect the target's availability or performance.

Red vs. Blue:
Red teams test the effectiveness of a security program or system by acting like attackers. Red teams are sometimes called tiger teams. Blue teams are defenders and may operate against red teams or actual attackers.
• Red team
A red team is an inside group that explicitly challenges a company's strategy, products, and preconceived notions. It frames a problem from the perspective of an adversary or sceptic, to find gaps in plans, and to avoid blunders, the red team simulates the hackers.
• Blue Team
A blue team is an inside group that works to defend a company's assets. Ideally, this is a group
of network security experts, they defend stuff from the hacking team.

Red Team Operations:
• Emulate the tactics of real-world threat actors
• Training of Blue Team / Incident Response staff
• Actively exercise the full incident response loop
• Gauge minimum time to detect, minimum time to recover
• Post-exploitation offensive data analysis

Different types of hackers:
•White hat—Hacks software primarily for benevolent purposes, such as security research, to find ways to improve software security.
•Blackhat—Hacks mainly for criminal purposes (such as extortion, theft, and cyberterrorism).
•Gray hat—Doesn't fit in the other two categories. Primarily motivated by profit, selling information they have uncovered to government agencies, for example.

Firewall:
• 1st generation: Packet filtering firewalls.
• 2nd generation: application (proxy) firewalls
• 3rd generation: state full packet firewalls
• 4th generation: dynamic filtering
• 5th generation: kernel proxy

Firewall Logs:
• Connections permitted or denied
• IDS activity
• Address translation audit trail
• User activity
• Cut-through-proxy activity
• Bandwidth usage
• Protocol usage

The PCI Data Security Standard goals:
• Build and Maintain a Secure Network
• Protect Cardholder Data
• Maintain a Vulnerability Management Program
• Implement Strong Access Control Measures
• Regularly Monitor and Test Networks
• Maintain an Information Security Policy
Note: PCI DSS allows for cardholder information at rest to be secured with either tokenization or encryption, but the use of one is mandatory.

Provide Secure Communication in Web

• Make sure the network is securely configured, but program as though network security will eventually be compromised.
• Use transport layer encryption (e.g., HTTPS, TLS, SSL) whenever possible—especially for communicating any sensitive data or session tokens to APIs and services.
• Protect data and requests transmitted between a client and server. For example, use parameterized queries to avoid Structured Query Language (SQL) injection.
• Do not send sensitive data over inappropriate channels, such SMS, MMS, or notification services.
• Account for outside entities (e.g., third-party analytics services, social networks) by using their SSL versions for routines run in the browser/webkit.
• Use strong, industry standard encryption algorithms with appropriate key lengths.
• Use certificates signed by a trusted CA provider, and use certificate pinning for security conscious applications.
• Require SSL chain verification, and establish a secure connection only after you verify the identity of the endpoint server using trusted certificates in the key chain.
• Fail safely, blocking communication and alerting the user if the application detects an invalid certificate.
• If practical, encrypt sensitive data before providing it to the SSL channel to provide an extra layer of defense in case the SSL/TLS layer is compromised.

Mobile devices are prime vectors for data loss; areas the professional should focus on:

• Secure communications
• Antimalware
• Strong authentication
• Passwords
• Control 3rd party software
• Separate secure mobile gateways
• Lockdown, audits
• Penetration tests
• Mobile security policy

Basic Types of Mobile Threats:

• Denial of service Deny or degrade service to users. Jamming of wireless communications, overloading networks with bogus traffic, ransomware, theft of mobile devices or mobile services.
• Geolocation Physical tracking of users. Passively or actively obtaining accurate three-dimensional coordinates of target, possibly including speed and direction.
• Information disclosure Unauthorized access to information or services.
Interception of data in transit, leakage or exfiltration of users, app, or enterprise data, tracking of user location, eavesdropping on voice or data communications, surreptitiously activating the phone's microphone or camera to spy on the user.
• Spoofing Impersonating something or someone. Email or SMS message pretending to be from the boss or colleague (social engineering); a fraudulent Wi-Fi access point or cellular base station mimicking a legitimate one.
• Tampering Modifying data, software, firmware, or hardware without authorization. Modifying data in transit, inserting tampered hardware or software into the supply chain, repackaging legitimate apps with malware, modifying network or device configuration (e.g., jailbreaking or rooting a phone).

Cybersecurity Framework:

• Identify – Develop the organizational understanding to manage cybersecurity risk to systems, assets, data, and capabilities. The activities in the Identify Function are foundational for effective use of the Framework. Understanding the business context, the resources that support critical functions, and the related cybersecurity risks enables an organization to focus and prioritize its efforts, consistent with its risk management strategy and business needs. Examples of outcome Categories within this Function include: Asset Management; Business Environment; Governance; Risk Assessment; and Risk Management Strategy.
• Protect – Develop and implement the appropriate safeguards to ensure delivery of critical infrastructure services. The Protect Function supports the ability to limit or contain the impact of a potential cybersecurity event. Examples of outcome Categories within this Function include: Access Control; Awareness and Training; Data Security; Information Protection Processes and Procedures; Maintenance; and Protective Technology.
• Detect – Develop and implement the appropriate activities to identify the occurrence of a cybersecurity event. The Detect Function enables timely discovery of cybersecurity events. Examples of outcome Categories within this Function include: Anomalies and Events; Security Continuous Monitoring; and Detection Processes.
• Respond – Develop and implement the appropriate activities to take action regarding a detected cybersecurity event. The Respond Function supports the ability to contain the impact of a potential cybersecurity event. Examples of outcome Categories within this Function include: Response Planning; Communications; Analysis; Mitigation; and Improvements.
• Recover – Develop and implement the appropriate activities to maintain plans for resilience and to restore any capabilities or services that were impaired due to a cybersecurity event. The Recover Function supports timely recovery to normal operations to reduce the impact from a cybersecurity event. Examples of outcome Categories within this Function include: Recovery Planning; Improvements; and Communications.

Attacks:

• Passive Attacks – hard to detect because the attacker is not affecting the protocol. Examples are Eavesdropping, network sniffing, and capturing data as it passes, used to gather data prior to an active attack.
• Active Attacks – Altering messages, modifying system files, and masquerading are examples because the attacker is actually doing something.
• Ciphertext Attacks - The attacker obtains ciphertext of several messages, with each message being encrypted using the same encryption algorithm. Attacker's goal is to discover the key. Most common attacks are easy to get ciphertext, but hardest attack to be successful at.
• Known-Plaintext Attack - The attacker has the ciphertext of several messages, but also the plaintext of those messages. The goal is to discover the key by reverse-engineering and trial/error attempts
• Chosen Plaintext Attack - The attacker not only has access to the ciphertext and associated plaintext for several messages, he also chooses the plaintext that gets encrypted. More powerful than a known-plaintext attack because the attacker can choose specific plaintext blocks to encrypt, ones that might yield more info about the key.
• Chosen-Ciphertext Attack: Attacker can choose different ciphertexts to be decrypted and has access to the decrypted plaintext. This is a harder attack to carry out, and the attacker would need to have control of the system that contains the cryptosystem
• Adaptive Attacks: Each of the attacks has a derivative with the word adaptive in front of it. This means that an attacker can carry out one of these attacks, and depend on what is gleaned from the first attack, the next attack can be modified. This is the process of reverse-engineering or cryptanalysis attacks.
• Birthday attack: a Cryptographic attack that exploits the math behind the birthday problem in the probability theory forces collisions within hashing functions.
• Brute force attacks: continually tries different inputs to achieve a predefined goal. Brute force is defined as "trying every possible combination until the correct one is identified".
• Buffer overflow: Too much data is put into the buffers that make up a stack. Common attacks vector are used by hackers to run malicious code on a target system.
• Cross-site scripting: refers to an attack where vulnerability is found on a website that allows an attacker to inject malicious code into a web application
• Dictionary attacks: Files of thousands of words are compared to the user's password until a match is found.
• DNS poisoning: Attacker makes a DNS server resolve a hostname into an incorrect IP address
• Fraggle attack: A DDoS attack type on a computer that floods the target system with a large amount of UDP echo traffic to IP broadcast addresses.
• Pharming: redirects a victim to a seemingly legitimate, yet fake, web site
• Phishing: type of social engineering with the goal of obtaining personal information, credentials, credit card number, or financial data. The attacker's lure, or fish, for sensitive data through various different methods
• Mail Bombing: This is an attack used to overwhelm mail servers and clients with unrequested e-mails. Using e-mail filtering and properly configuring email relay functionality on mail servers can be used to protect this attack.
• Ping of Death: A DoS attack type on a computer that involves sending malformed or oversized ICMP packets to a target.
• Replay attack: a form of network attack in which a valid data transmission is maliciously or fraudulently repeated with the goal of obtaining unauthorized access.
• Replay Attack: an attacker capturing the traffic from a legitimate session and replaying it to authenticate his session
• Session hijacking: If an attacker can correctly predict the TCP sequence numbers that the two systems will use, then she can create packets containing those numbers and fool the receiving system into thinking that the packets are coming from the authorized sending system. She can then take over the TCP connection between the two systems.
• Side-channel attacks: Nonintrusive and are used to uncover sensitive information about how a component works, without trying to compromise any type of flaw or Weakness. A noninvasive attack is one in which the attacker watches how something works and how it reacts to different situations instead of trying to "invade" it with more intrusive measures. side-channel attacks are fault generation, differential power analysis, electromagnetic analysis, timing, and software attacks.
• Smurf attack: A DDoS attack type on a computer that floods the target system with spoofed broadcast ICMP packets.

• Social engineering: An attacker falsely convinces an individual that she has the necessary authorization to access specific resources.
• Spoofing at Login: an attacker can use a program that presents the user with a fake login screen, which often tricks the user into attempting to log on
• SYN flood: DoS attack where an attacker sends a succession of SYN packets with the goal of overwhelming the victim system so that it is unresponsive to legitimate traffic.
• TOC/TOU attack: Attacker manipulates the "condition check" step and the "use" step within the software to allow for unauthorized activity.
• War dialing: war dialer inserts a long list of phone numbers into war dialing program in hopes of finding a modem to gain unauthorized access.
• Wormhole attack: This takes place when an attacker captures packets at one location in the network and tunnels them to another location in the network for a second attacker to use against a target system.
• Denial-Of-Service (Dos) Attack: An attacker sends multiple service requests to the victim's computer until they eventually overwhelm the system, causing it to freeze, reboot, and ultimately not be able to carry out regular tasks.
• Man-In-The-Middle Attack: An intruder injects herself into an ongoing dialog between two computers so she can intercept and read messages being passed back and forth. These attacks can be countered with digital signatures and mutual authentication techniques.
• Teardrop: This attack sends malformed fragmented packets to a victim. The victim's system usually cannot reassemble the packets correctly and freezes as a result. Countersues to this attack is to patch the system and use ingress filtering to detect these packet types.

Guidelines to Prevent Buffer Overflow Defects:
• Validates user input for type and length to ensure it will not overflow the legitimate data boundaries.
• Uses the least privileges possible for the accounts in which your processes run.
• Is especially careful when passing input parameters to other code, especially unmanaged code, DLLs, etc.
• (If you use third-party libraries) uses only libraries that you have researched very carefully to ensure they are free from buffer overflow vulnerabilities.

Prevent Buffer Overread Defects:
• Make sure that the start location for each read operation remains within the buffer boundaries.
• Make sure that the end location for each read operation remains within the buffer boundaries.

Prevent Uncontrolled Format String Defects:
• You heed compiler messages that warn about potential format string problems.
• Your software does not create format strings from user input.

Prevent Race Conditions:
To prevent race conditions:
• Lock the shared resource when the process is modifying it, and unlock it when the process is done. Note that this approach can get quite complicated.
• Example: An error occurs in the process that has locked the resource, so you will deadlock other processes that use that resource. In some cases, it may be preferable to leave the resource locked than to continue in an undefined state.
• Write code that doesn't depend on side effects. A side effect is when a function changes a variable outside of its own scope. If possible, only modify data inside of the local scope of the thread or process.
• Write temporary files in a data store that is available only to a single thread or process.
• Research best practices for writing multiple threading or multiple processing code in the language and system you are developing in. Tools you might use (if available) include semaphores, mutexes, and others.

Sandboxing:
A sandbox is an isolated environment in which untrusted data can be tested; Allows for analysis of applications and data in a secure environment without risk to the production environment.

Hardening Best Practices:
- Defense in depth: Use the tools as an additional layer of defense.
- Access control: Tightly control and monitor access to these tools.
- Auditing and monitoring: Track and validate usage of the tools.
- Maintenance: Update tools as necessary and follow vendor recommendations

Review of Network Security Controls:
- Vulnerability assessments
- Network security groups (access lists)
- VLANs
- Access control
- Use of secure protocols, such as TLS and IPSec
- IDP/IPS systems
- Firewalls
- Honeypots
- Zoning of storage traffic (like VLANs for storage data)
- Vendor-specific security products (VMware vCloud Networking and Security or NSX products)
- Keep public data and private data on separate virtual switches

Wireless Attack:
- Rogue AP
- Interference
- Jamming
- Evil Twin
- War Driving
- War Chalking
- IV attack
- WEP/WPA attacks

Secure configuration of Hardware devices:
- Secure build
- Secure initial configuration
- Host hardening - remove all non-needed
- Host Patching
- Host lockdown
- Secure ongoing configuration, maintenance

RFID Attacks:
- RFID Counterfeiting
- RFID Sniffing
- Tracking
- Denial of Service
- Spoofing
- Repudiation
- Insert Attacks
- Replay Attacks
- Physical Attacks
- Viruses

RFID attacks:
- Eavesdropping/Skimming
- Traffic Analysis
- Spoofing
- Denial of Service Attack/Distributed Denial of Service Attack
- RFID Reader Integrity
- Personal Privacy

Attacks on VLAN:
- MAC Flooding Attack
- 802.1Q and Inter-Switch Link Protocol (ISL) Tagging Attack
- Double-Encapsulated 802.1Q/Nested VLAN Attack
- ARP Attacks
- Multicast Brute Force Attack
- Spanning-Tree Attack
- Random Frame Stress Attack

Methods of Cryptanalytic Attacks:
- Cipher text-Only Attack (Only Ciphertext)
- Known Plaintext (Both Plaintext and Ciphertext available)
- Chosen Plaintext (Known algorithm, Adaptive where Plaintext can be changed)
- Chosen Ciphertext (Known algorithm, Adaptive where Ciphertext can be changed)

Common vulnerabilities and threats of Security Architecture:
- Poor memory management
- Covert channels (storage and timing)
- Insufficient system redundancy
- Poor access control
- Hardware failure
- Misuse of privileges
- Buffer overflows
- Memory attacks
- DoS
- Reverse engineering
- Hacking
- Emanations
- State attacks (race conditions)

A honeypot can be used:
- Gathering threat intelligence
- Distracting attackers
- Delaying attackers

Endpoint Protection:
- Built-in firewall functionality.
- Intrusion detection system (IDS) /intrusion prevention system (IPS) functionality.
- Data loss prevention (DLP) functionality.
- Application whitelisting / blacklisting functionality.
- Full disk encryption.
- Management interfaces for configuration of each endpoint or groups of endpoints.
- A centralized in-house server for distributing malware signature updates.

Note: A discovery tool is a primary component of a DLP solution. This might be employed for purposes of identifying and collecting pertinent data.

DLP Architecture
- Data in motion (DIM): Network-based or gateway DLP. Monitors SMTP, HTTP, HTTPS, SSH, FTP, etc., for sensitive data and prevents it from leaving the organization.
- Data at rest (DAR): Storage-based. Used for tracking and identification data as it's installed on the system where the data resides. Generally needs another mechanism for any enforcement.
- Data in use (DIU): Client- or endpoint-based. Resides on users' workstations. Requires great amount of management. Not easy to deploy and manage.

DLP Policy Considerations:
- What classification of data is permitted to be stored in the cloud?
- Where can this data be stored (geographical locations)?
- How should this data be stored (encrypted)?
- What type of access controls need put in place?
- Who, what, where, when, can data be accessed by or from?
- When can data leave the cloud, if ever?

Block Storage:
- Primary role of storage is to group disks together into logical volumes (LUNs, virtual disks, generic volume storage, and elastic block storage)
- None of these have a file system when created
- It's up to the OS on the VM to create the file system

Object Storage:

- Has a flat file system on it
- Provides for simple object storage (files of nearly any type)
- Objects are accessible via browser and REST API
- AWS refers to these as buckets in their S3 service
- Rackspace offers cloud files
- Object storage is typically the best way to store an OS (image)
- Data can be replicated across multiple object storage servers or sites
- Offer basic file usage, nothing fancy

General types of viruses:

• File Infectors – Infects program or object files.
• Boot sector infectors – Attach or replace boot records
• System Infectors – Attaches to system files or system structure
• Companion virus – Does not physically touch the target file
• Email Virus – Aware of the email system.
• Multipartite – Reproduces in more than one way
• Macro Virus – Uses macro programming of the app. Infect data files
• Script Virus – Standalone files that can be executed by an interpreter
• Script host – .vbs as host to script virus.

RAID:

Some of the RAID protection options are:
• RAID0 – Striped
• RAID 1 withstands failure of one drive within one of the mirrored pairs. The number of required drives is twice the amount required to store data.
• RAID2 - Hamming Code requires either 14 or 39 disks
• RAID3 - Striped Set with Dedicated Parity (Byte Level)
• RAID4 - Striped Set with Dedicated Parity (Block Level)
• RAID 5 protection is also available. Data blocks are striped horizontally across the members of a RAID 5 group, and each member owns some data tracks and some parity tracks.
• RAID 6 protects data with failures of up to 2 drives per RAID group.
• RAID1+0 - striped set of mirrored disks

Power:

• Blackout: Generator
• Brownout: (UPS) Uninterruptible Power Supply
• Surge: Surge protector
• Spike: Surge protector
• Noise: Power conditioner
• Clean power: No solution is needed

Hashing:

• MDS Message-Digest Algorithm - 128-bit digest
• SHA - 160-bit digest
• HAVAL
• RIPEMD-160
• Birthday attacks possible

Symmetric Algorithms:

• Data Encryption Standard (DES)
• 3DES (Triple DES)
• Blowfish
• Twofish
• International Data Encryption Algorithm (IDEA)
• RC4, RCS, and RCG
• Advanced Encryption Standard (AES)
• Secure and Fast Encryption Routine (SAFER)
• Serpent
• CAST

Asymmetric Algorithms:

• RSA - factoring the product of two large prime numbers
• Diffie-Hellmann Algorithm
• EI Gamal- discrete logs
• Elliptic Curve Cryptography (ECC)

Encryption:
• Use encryption that is strong enough to protect the data.
• But the stronger encryption is, the longer it will take to decrypt.
• Whatever encryption you use, it shouldn't slow down performance unacceptably for most of your users.

Other uses for encryption include:
• Non-repudiation • Digital Rights Management (DRM) • Digital Signature • Tunneling

Certificate Revocation:
•Certificates revoked when:
•• They expire.
•• Security of private key is in doubt.
•CRL
•• List issued periodically by CA of certificate serial numbers that have been revoked.
•• Provides reasons for revocation.
•• CRL has a digital signature to prevent spoofing or DoS attacks.
•• List has a short lifetime.
•OCSP
•• Uses HTTP request to obtain revocation status from CA.
•• Provides faster confirmation than CRL.

Cryptography:
• Privacy
• Authentication
• Integrity
• Non-repudiation

Security Concepts:
• Need-to-Know (access only to what's needed to perform task/job).
• Separation of Duties (one person cannot execute all steps of critical processes or engage in a malicious activity without collusion).
• Monitor special privileges (audit logs for system operators /administrators/data center employees ensure privileged users cannot circumvent security policy, should not have access to their logged activity, conduct background investigations).
• Job rotation (reduces collusion).
• Information lifecycle: (creation, use, destruction of data, information/data owner helps safeguard data by classifying and determining its criticality and sensitivity).

Hashing:
• Provides a way to hide sensitive data
• Allows for an integrity check of the data by checking it against the hashed value
• The hashed value in no way can be used to identify the original data

Masking and Obfuscation:
• Data obfuscation is the process of changing data so it doesn't appear to be what it is.
• Generally used to comply with standards by masking sensitive data (SSN, DOB, etc.).
• Sometimes used to turn production data into testing data by masking sensitive data points.

Black/White List:
• The blacklist is an explicit deny.
• The whitelist is an implicit deny.
• The blacklist = "If you are on the list, then you are NOT allowed in."
• The whitelist = "If you are NOT on the list, then you are NOT allowed in."

Client-based vulnerabilities, Client system should have:
• Licensed as running
• Current antivirus and antimalware
• HIDS
• Strong encryption
• Limited accounts without administrative privileges
• Continuous monitoring
• Hardened mobile devices

Server-based vulnerabilities, Server system should:
• Determine how remote access will be established
• Check configuration management be performed
• Control data flow

Methods for defeating a switch:
• MAC Spoofing Set the MAC address of a NIC to the same value as another
• MAC Flooding Overwhelm the CAM table of the switch so it coverts to hub mode
• ARP Poisoning Inject incorrect information into the ARP caches of two or more endpoints.

Most important elements that record state data on network devices:
• Routing tables
• CAM tables
• NAT tables
• DNS cache
• ARP cache

Logical Security:
• Fail Open/Soft (availability is preserved, but data may not be secure)
• Fail Secure/Closed (data is secure, but availability is not preserved) Physical Security
• Fail Safe/Open (systems are shut down / entrances unlocked - humans are safe)
• Fail Secure/Closed (entrances are locked)
• Failover is a fault tolerance (redundancy) concept. If you have two redundant NICs; a primary and a backup – and the primary fails, the backup is used.

Database Model should provide:
• Transaction persistence
• Fault tolerance/recovery
• Sharing
• Security controls

Threats to a DBMS include:
• Aggregation (combining data to form sensitive information)
• Bypass attacks (avoiding controls to access information)
• Compromising database views (modifying/accessing restricted views)
• Concurrency (processes running at the same time without proper locks)
• Contamination (corruption)
• Deadlocking (denying users who access information at the same time)
• DoS (preventing authorized access)
• Improper modification (accidental/intentional)
• Inference (deducing restricted information by observation)
• Interception of data
• Server access
• Polymorphism
• Polyinstantiation
• TOC/TOU (malicious changing data at a certain time)
• Web security issues
• Unauthorized access

Aggregation vs. Inference:
Inference (understand business, risk analysis, interview owner); by combining multiple reports or source of information, you succeed in guessing or making up new information. Aggregation (understand data and fields); the sum may represent a level of security higher than each of the parts. Be aware of these terms:
• Polyinstantiation: Prevents inference attacks
• Database Views: Constrained interfaces, restrictive interface
• Context-dependent access control: Content dependent controls
• Noise and perturbation: Addresses inference attacks
• Cell suppression: A technique used against the inference
Noise and perturbation: A technique of inserting bogus information in the hopes of misdirecting an attacker or confusing the matter enough that the actual attack will not be fruitful.

Tokens - "Synchronous" vs. "Asynchronous":
• Synchronous Dynamic Password Tokens Hardware tokens that create synchronous dynamic passwords are time-based and synchronized with an authentication server. They generate a new password periodically, such as every 60 seconds. This does require the token and the server to have accurate time.
• Asynchronous Dynamic Password Tokens does not use a clock. Instead, the hardware token generates passwords based on an algorithm and an incrementing counter. When using an incrementing counter, it creates a dynamic one-time password that stays the same until used for authentication. Some tokens create a one-time password when the user enters a PIN provided by the authentication server into the token.

Token Usage:
• Single-token authentication
• Multi-token authentication

Types of tokens for e-authentication:
• Memorized Secret Token
• Pre-registered Knowledge Token
• Look-up Secret Token
• Out of Band Token
• Single-factor (SF) One-Time Password (OTP) Device
• Single-factor (SF) Cryptographic Device
• Multi-factor (MF) Software Cryptographic Token
• Multi-factor (MF) One-Time Password (OTP) Device
• Multi-factor (MF) Cryptographic Device

Token Threats:
• Something you have may be lost, damaged, stolen from the owner or cloned by the Attacker.
• Something you know may be disclosed to an Attacker. Attacker might guess a password/PIN.
• Something you are may be replicated.

Token Threat Mitigation Strategies:
• Multiple factors make successful attacks more difficult to accomplish.
• Physical security mechanisms may be employed to protect a stolen token from duplication.
• Imposing password complexity rules may reduce the likelihood of a successful guessing attack.
• System and network security controls may be employed to prevent an Attacker from gaining access to a system or installing malicious software.
• Periodic training may be performed to ensure the Subscriber understands when and how to report compromise (or suspicion of compromise) or otherwise recognize patterns of behavior that may signify an Attacker attempting to compromise the token.
• Out of band techniques may be employed to verify proof of possession of registered devices (e.g., cell phones).

Token Threat/Attack:
• Theft - Use multi-factor tokens which need to be activated through a PIN or biometric.
• Duplication - Use tokens that are difficult to duplicate, such as hardware cryptographic tokens.
• Discovery - Use methods in which the responses to prompts cannot be easily discovered.
• Eavesdropping
•• Use tokens with dynamic authenticators where knowledge of one authenticator
does not assist in deriving a subsequent authenticator.
•• Use tokens that generate authenticators based on a token input value.
•• Establish tokens through a separate channel.
• Offline cracking
•• Use a token with a high entropy token secret
•• Use a token that locks up after a number of repeated failed activation attempts.
• Phishing or pharming - Use tokens with dynamic authenticators where knowledge of one authenticator does not assist in deriving a subsequent authenticator.
• Social engineering - Use tokens with dynamic authenticators where knowledge of one authenticator does not assist in deriving a subsequent authenticator.
• Online guessing - Use tokens that generate high entropy authenticators.

Key States and Transitions:
• The pre-activation state: The key has been generated, but not yet authorized for use
• The active state: The key may be used to cryptographically protect information
• The deactivated state: The crypto period of the key is expired, but the key still needs to perform cryptographic operations
• The destroyed state: The key is destroyed here
• The compromised state: The key is released or determined by an unauthorized entity
• The destroyed compromised state: The key is destroyed after a compromise or the compromise is found after the key is destroyed

Key Management:
• Secure generation of keys
• Secure storage of keys
• Secure distribution of keys
• Secure destruction of keys

Secure Key Management:
• Key Generation: How, when, and on what device keys are generated
• Key Derivation Constructing cryptographic keys from other keys and variables
• Key Establishment: Two parties algorithmic computation of keying material
Secure wrapping and sending keys from one device to another
• Key Storage: Secure storage of keys (frequently encrypted using 'key encryption keys') and in what type of device(s)
• Key Lifetime: How long a key should be used before being destroyed (zeroized)
• Key Zeroization: the Secure destruction of key material
• Accounting: Identifying, tracking and accounting for the generation, distribution, and destruction of key material between entities

Key Management Factors:
• Key control measures: Determine who has access to keys and how they are assigned.
• Key recovery: How lost keys are recovered.
• Key storage: A secure repository for key assignment records.
• Key retirement/destruction: How keys are removed from use and how they are destroyed.
• Key change: How keys are changed on a periodic basis.
• Key generation: How keys are generated to ensure they are random.
• Key theft: What to do when keys have been compromised.
• The frequency of key use: How to limit the time that keys are used and frequency of key reuse.
• Key escrow—Provides law enforcement and other agencies authorized access to encrypted information. Keys may have to be stored at different locations

Project Management Quick Reference:
• The work package is the LOWEST level on a WBS.
• The WBS doesn't show the order of the work packages or any dependencies between them.
• WBS Dictionary – Detailed description of the WBS component
• Cost Benefit: Looking at how much your quality activities will cost
• Stakeholders are ONLY the interested entities that are internal or external to the organization.
• Project life cycle approach is Project governance and is described in the project management plan.
• Risk and uncertainty are greatest at the start of the project.
• Analysis of project forecasts (including time and cost) is also part of Performance Reporting.
• Risk appetite is the degree of uncertainty an entity is willing to take on in anticipation of a reward.
• Risk tolerance is the degree, amount, or volume of risk that an organization or individual will withstand.
• Risk threshold refers to measures along the level of uncertainty or the level of impact at which a stakeholder may have a specific interest.
• Positive and negative risks are commonly referred to as opportunities and threats.
• Project risk could exist at the moment a project is initiated.
• The procurement SOW describes the prospective sellers if they are capable of providing the products, services, or results.
• PMO manages the methodologies, standards, overall risks/opportunities, metrics, and interdependencies between projects at the enterprise level. Supportive, Controlling and Directive are the types of PMO structures in organizations.
• UNILATERAL: this is a special class of contract in which the seller doesn't have to explicitly accept the offer in order for a contract to be established. This is a unilateral contract, and the best example is a purchase order (PO)
• Force Majeure Risks, such as Earthquakes, Floods, Acts of Terrorism, Etc., should be covered under Disaster Recovery Procedures instead of Risk Management.

Quality of Service Metrics:
• Availability
• Outage Duration
• Mean Time Between Failures (MTBF)
• Capacity Metric
• Performance Metrics
• Reliability Percentage Metric
• Storage Device Capacity Metric
• Server Capacity Metric
• Instance Startup Time Metric
• Response Time Metric
• Completion Time Metric
• Mean Time to Switchover Metric
• Mean Time System Recovery Metric
• Scalability Component Metrics
• Storage Scalability Metric
• Server Scalability Metric

Identity and Access Management (IAM) Lifecycle:
• Provisioning: Applying appropriate rights to users for files/folders
• Review: Periodic monitoring of existing rights for the continued need
• Revocation: Removal of rights when no longer needed warranted

Phases of IAM:
• Provisioning and de-provisioning
• Centralized directory services
• Privileged user management
• Authentication and access management

Key issues with Identity Services:
• APIs: While IAM vendors offer connectors to the most common cloud services, they are unlikely to provide all the connectors you need.
• Authorization Mapping: There are many possible ways to specify authorization rules, such as by role vs. by attribute.
• Audit: In-house systems can be linked with log management and SIEM systems to produce compliance reports and provide monitoring and detection of security events.
• Privacy: Users, user attributes, and other information are often pushed outside your corporate network and into one or more cloud data repositories.
• Latency: Propagating rule changes from internal IAM to cloud IAM can take some time. Latency is a subject to discuss with both your IAM provider and cloud service provider.
• Privileged User Management: This has been a problem for a long time, and the cloud adds a new wrinkle. Historically privileged users were all employees, and if things went pear-shaped, you could handle it as an HR event. In the cloud that breaks down.
• App Identity: Once you have the user logged in you might still need to verify the application they are using — or perhaps there is no user at all, just middleware.
• Mobile: mobile connections to cloud services occur outside of the boundaries of normal.
• Identity Store Location: If companies are moving their applications and data to cloud services, will they also move existing identity stores?

A comprehensive and effective security intelligence process can produce:
• Faster detection and remediation of threats.
• Improved regulatory compliance.
• Reduction of fraud, theft, and data leakage.
• Reduction of effort needed to provide security and deal with fallout related to breaches.
• The ability to detect potential weaknesses before an exploit actually occurs.

Security Intelligence Collection Lifecycle:
• Planning and direction
• Collection
• Processing
• Analysis and production
• Dissemination and integration

Cloud Service Models:
• Software as a Service (SaaS)
  •• Provider's applications run in the cloud
  •• Clients use thin apps (like a browser) to access SaaS
• Platform as a Service (PaaS)
  •• Client apps deployed into and running in the cloud
• Infrastructure as a Service (IaaS)
  •• Processing, storage, and network services
  •• Client controls operating systems and host configurations

Note: You remain accountable and responsible – regardless of any cloud service used.

Outsourcing:
• Ensuring that the organization has appropriate controls and processes in place to facilitate outsourcing.
• Ensuring that there are appropriate information risk management clauses in the outsourcing contract.
• Ensuring that a risk assessment is performed for the process to be outsourced.
• Ensuring that an appropriate level of due diligence is performed prior to contract signature.
• Managing the information risk for outsourced services on a day to day basis
• Ensuring that material changes to the relationship are flagged and new risk assessments are performed as required.
• Ensuring that proper processes are followed when relationships are ended.

Contracts with third parties include:
• Agreement that the vendor will comply with applicable information security and privacy laws and regulations.
• Information security and privacy safeguards.
• Right-to-audit
• Notification in the event of a data breach.
• Where the data will be accessed, stored, and/or processed. It is important to know the specific locations and ensure that the vendor will notify the primary entity if there is a need to add, change, or remove a location.
• Data return or destruction when a contract terminates.
• Employee background checks/employment verification.
• Expectations for employee training.
• The ability of the vendor to subcontract work.
• Business continuity/disaster recovery plans. Within what time frame must the vendor's function be operational in the event of a disaster?

Third Party Contracts:
• NDA/NDC
• Regulatory Compliance
• Incident notification
• SLA/SLC

Evaluate the Third party:
• On-Site Assessment
• Document Exchange and Review
• Process/Policy Review

Popular services:
• IaaS: Amazon EC2, Windows Azure, Rackspace (backup)
• PaaS: Google App Engine, Cloud Foundry, force.com
• SaaS: Office 365, Dropbox, salesforce.com, Google Apps
• Cloud management: CloudStack, OpenStack

Evaluating Cloud Service Security:
• What is the security of the facility running the servers?
• Is client data encrypted? If so, what encryption method is being used?
• Is the cloud provider's internal system segregated from its internet-facing cloud servers?
• Does the provider have a security audit they can share with us?
• What safeguards do they employ on their web service interface and/or API?
• Do they back up their data regularly and perform test restores for proper disaster recovery?
• What general data breach and protection policies are in place?
• Is client data shared with any third parties?

Data Retention Policy in Cloud:
• Regulation
• Data mapping
• Data Classification
• Procedures
• Monitoring and maintenance

The Cloud Secure (SDLC):
- Defining
- Designing
- Development
- Testing
- Secure Operations
- Disposal

Cloud computing impacts four areas of Governance and Risk Management:
- Governance includes the policy, process, and internal controls that comprise how an organization is run.
- Enterprise risk management includes managing overall risk for the organization, aligned with the organization's governance and risk tolerance.
- Information risk management covers managing the risk to information, including information technology.
- Information security is the tools and practices to manage risk to information.

Cloud security – general areas of concern:
- Governance and Enterprise Risk Management
- Legal Issues: Contracts and Electronic Discovery
- Compliance and Audit
- Information Management and Data Security
- Portability and Interoperability
- Traditional Security, Business Continuity and Disaster Recovery
- Data Center Operations
- Incident Response, Notification and Remediation
- Application Security
- Encryption and Key Management
- Identity and Access Management
- Virtualization
- Security as a Service

ENISA Cloud Security Document:
- LOSS OF GOVERNANCE; CSP does not commit to the necessary task
- VENDOR LOCK-IN, the high cost of moving to a different vendor
- ISOLATION FAILURE: one tenant influences another.
- COMPLIANCE RISKS: i.e. Audit impossible, or no evidence
- MANAGEMENT INTERFACE COMPROMISE
- DATA PROTECTION; protection cannot be demonstrated
- INSECURE OR INCOMPLETE DATA DELETION
- MALICIOUS INSIDER: i.e. Cloud provider or auditor

Cloud Storage Security:
- Encryption
- Authentication
- Authorization

Security in Cloud Computing:
- Data segregation
- Identity Management
- Availability Management
- Vulnerability Management
- Access Control Management

Steps to take on the cloud to avoid vendor lock-in:
- Do your due diligence
- Plan early for an exit
- Design your application to be loosely coupled
- Maximize portability of your data
- Consider a multi-cloud strategy
- Implement DevOps tools and processes

Note: A poorly crafted contract can lead to vendor lock-in

Critical issues to cloud security:
- Data Breaches
- Weak Identity, Credential, and Access Management
- Insecure APIs
- System and Application Vulnerabilities
- Account Hijacking
- Malicious Insiders
- Advanced Persistent Threats (APTs)
- Data Loss
- Insufficient Due Diligence
- Abuse and Nefarious Use of Cloud Services
- Denial of Service
- Shared Technology Issues

Cloud Risk:
- Privileged user access
- Regulatory compliance
- Data Location
- Data Segregation
- Recovery
- Long-term viability

SLA in Cloud:
- Availability
- Performance (e.g. Maximum response times)
- Security/privacy of the data (e.g. Encrypting all stored and transmitted data)
- Disaster Recovery expectations (e.g. Worse case recovery commitment)
- Location of the data (e.g. Consistent with local legislation)
- Access to the data (e.g. Data retrievable from a provider in readable format)
- Portability of the data (e.g. Ability to move data to a different provider)
- The process to identify problems and resolution expectations
- Change Management process (e.g. Changes – updates or new services)
- Dispute mediation process (e.g. escalation process, consequences)
- Exit Strategy with the expectations of the provider to ensure a smooth transition

Preparing for Cloud Use:
- Framework for Cloud Governance
- Planning for Cloud use
- Security controls for Cloud use
- Security Awareness Training for Cloud Users
- Performing due diligence on intended Cloud Service Providers (CSPs)

The CSP Agreement:
- Required services, service levels, uptime, redundancy, recovery
- Confidentiality / Non-Disclosure / Ownership / Access
- Compliance guarantees with notification and penalties for violations
- Breach / Incident detection, notification, response, and remediation
- Prudent management of the CSP business
- Monitoring, auditing, inspections, maintaining metrics, reports

Essential characteristics of Cloud:
- Resource pooling. Multiple customers
- On-demand self-service. Unilateral provisioning
- Broad network access. Network and client
- Rapid elasticity. Speedy provisioning and deprovisioning
- Measured Service. Pay per use

Cloud Data Life Cycle:
- Create: Creation is the generation of new digital
- Store: Storing is the act committing the digital data
- Use: Data is viewed, processed, or otherwise used
- Share: Information is made accessible to others
- Archive: Data leaves active use and enters long-term storage
- Destroy: Data is permanently destroyed

Identity as a Service IDaaS:

Identity as a Service (IDaaS) is an authentication infrastructure that is built, hosted and managed by a third-party service provider. IDaaS can be thought of as single sign-on (SSO) for the cloud. This can provide benefits including integration with cloud services and remove overhead for maintenance of traditional on-premise identity systems, but it can also create risk due to the third-party control of identity services and reliance on an offsite identity infrastructure. An IDaaS solution via a cloud provider usually includes the following:
- Single sign-on
- Provisioning
- Password management
- Access governance

Cloud:
- Anything as a Service (XaaS): The understanding that there is a vast amount of services available across the internet so you don't need to stand up an on-premises solution.
- Cloud app: A cloud application accessed across the internet, not installed locally. Cloud Application Management Platform (CAMP): A specification designed to ease management of applications across public and private cloud platforms.
- Cloud computing: A type of computing that shares computing resources of remote environments to accomplish work, instead of using local servers.
- Cloud database: A database accessible to clients across the internet. Also refers to Database as a Service (DBaaS). These cloud databases use cloud computing to achieve optimization, scaling, high availability, and multi-tenancy.
- Cloud enablement: Making cloud services available to a client.
- Cloud management: Software and technology used to monitor and operate cloud environments. These tools help ensure cloud resources are working optimally.
- Cloud portability: The ability to move applications and data between different cloud service providers (CSPs) or between public and private cloud environments.
- Cloud provisioning: The deployment of cloud services to meet a need.
- Cloud service provider (CSP): Company providing cloud services to customers.
- Desktop as a Service (DaaS): A virtual desktop infrastructure (VDI), also called a hosted desktop service. Simply a desktop in the cloud you connect to and use with applications installed on it.
- Enterprise application: Applications or software used by large organizations. Generally refers to large-scale applications not suited for small business or individual needs.
- Hybrid cloud storage: A combination of public and private cloud storage. Sensitive data will reside in the private cloud, while other data or applications may reside in the public cloud.
- Infrastructure as a Service (IaaS): A computer infrastructure being delivered as a service. This includes compute, storage, network, and internet access. Simply a fully functional virtual environment on which customers provision virtual hosts.
- Managed service provider (MSP): Managed service providers (MSPs) provide various IT services to customers such as monitoring, patching, help desk, and network operations center.
- Multi-tenant: Having multiple customers using the same public cloud. Data is logically separated by security controls but still runs on the same shared underlying hardware.
- On-demand services: Service model allowing customers to scale their consumed resources without assistance from the provider in real time.
- Platform as a Service (PaaS): A cloud-based platform on which clients deploy their applications. The CSP will manage all underlying infrastructure, including operating system, compute hardware, and network. The customer is only responsible for managing their application code.
- Private cloud: Also known as internal or corporate cloud, this cloud compute platform is protected by the corporate firewall under the control of the IT department, not a CSP. Allows the IT department to control the security of the data and meet regulatory compliance. May be a corporate-owned data center referred to as a corporate cloud or can also be provided by a CSP offering isolated cloud services.
- Public cloud storage: Cloud storage in which the enterprise and storage services provider are separate and the data is stored outside the enterprise data center.
- Software as a Service (SaaS): Cloud-based software offered to clients accessed across the internet, most often as a web-based service. Think web-based applications you log in to and use.
- Vertical cloud computing: The optimization of cloud computing and services for a specific industry. An example would be cloud resources for the entertainment industry, which may have GPUs available to increase compute power for rendering of images and video.

Cloud API Security Concern:

A cloud API is basically used to integrate applications in order to enhance the cloud experience and provide inter-cloud compatibility. They are broadly classified into two categories: in-process APIs and remote APIs.
- Ensuring proper security measures to safeguard hypervisor to any sort of security threat.
- Careful assessment of the security practices as implemented by the cloud service providers need to be done before adopting any of them
- Proper SLAs between the customer and the CSP, defining the organizations' security requirements that need to be addressed.
- APIs in use need to be looked after and screened carefully. In the current scenario, most of the organizations prefer an integration of security techniques with their service models. They should be aware of the security implications associated with the usage of these cloud services. Reliance on weak APIs may jeopardize the security of important organizational data.

Provide Secure Password Management:
• Send non-temporary passwords only over an encrypted connection or as encrypted data (such as in an encrypted email). Temporary passwords associated with email resets may be an exception.
• Enforce password complexity requirements established by policy or regulation.
• Prevent password re-use.
• Hide password entry on the user's screen by default.
• Disable accounts after an established number of failed login attempts to prevent brute force attacks.
• Require the same level of security controls for password reset and changing operations as you require for account creation and authentication.
• Support sufficiently random answers for password reset questions.
• If using email-based resets, only send email to a pre-registered address with a temporary link/password.
• Provide a short expiration time for temporary passwords and links.
• Require temporary passwords to be changed on the next use.
• Notify users when their password has been reset (outside of the application, using their preregistered email address, for example).
• Enforce password changes based on requirements established in policies and regulations.
• Change all default passwords and user IDs provided with the development platform or services.

Protect Data in Transit and at Rest:
• Minimize the need for encryption by eliminating collection and storage of sensitive data as much as possible.
• Base your selection of cryptographic and key management algorithms to use on the objectives of the application, and use the most appropriate algorithm suite for the objectives. (Don't default to simply using libraries that are already available to you.)
• Implement all cryptographic functions on a trusted system, such as a server.
• Generate random numbers, random file names, random GUIDs, and random strings using your encryption library's approved random number generator.
• Use encryption libraries that comply with FIPS 140-2 or an equivalent standard.
• Ensure that cryptographic modules fail securely.
• Utilize a single standard TLS implementation that is configured appropriately.
• Use encryption (such as provided by TLS) to protect all sensitive information (such as credentials) sent over the network.
• Pre-encrypt files you must transmit over unencrypted channels.
• When encrypted channels fail, do not fall back to an unsecure connection, unless you encrypt data before sending.
• Make sure TLS certificates are valid and have the correct domain name, not expired, and installed with intermediate certificates when required.
• Specify character encodings for all connections.
• Protect all sensitive data stored on the server (include caches and temporary copies) from unauthorized access.
• Remove any sensitive data from the system as soon as it is no longer required.
• Purge temporary working files as soon as they are no longer needed.
• Use strong encryption algorithms.
• Protect server-side source-code from being downloaded by a user.
• Do not store passwords, connection strings, or other sensitive information in cleartext on the client side.
• Disable auto-complete features on forms expected to contain sensitive information, including authentication.
• Disable client-side caching on pages containing sensitive information.
• Establish and follow a policy and process for managing encryption keys.

Cloud services are broken down into three capabilities:
- Application capability: The cloud service customer uses the cloud service provider's applications.
- Infrastructure capability: The cloud service customer can provision and use processing, storage, or networking resources.
- Platform capability: The cloud service customer can deploy, manage, and run their own applications using one or more programming languages and one or more execution environments supported by the CSP.

Cloud deployment model:
The cloud deployment model you select will be based on:
- Risk appetite
- Cost
- Compliance and regulatory requirements
- Legal obligations
- Business strategy

Cloud Computing Roles
Cloud customer: An individual or organization that uses cloud-based services. Cloud service auditor: A third party that verifies CSPs are meeting service-level agreements (SLAs). Cloud service brokerage (CSB): Organization that looks to add value to cloud services through relationships with multiple CSPs. They are used to help customers identify the best cloud solution for them. CSBs sometimes resell cloud services. Cloud service provider: Company providing cloud services to customers. Cloud service partner: Includes other roles, such as cloud service auditor and cloud service broker.

CapEx vs. OpEx:
- CapEx:
  Capital expenditure (CapEx) is an upfront investment of a sum of money into a business requirement, such as a building, server farm, network environment, or network operations center (NOC)
- OpEx:
  An OpEx is an operational expenditure where you are paying for a service on a schedule. An example of this would be a building lease or utilities. However, this applies to services as well, such cloud services or hosting services
- CapEx vs. OpEx:
  Businesses may not have large amounts of capital at their disposal, so instead of building out an on-premises compute solution that could cost large amounts of money, they may instead look into a cloud solution. If the business stands up their compute needs in the cloud, they are paying month to month on only what is needed. Whereas, if they built out their on-premises solution, they would have to invest large amounts of capital upfront and then pay for ongoing costs, such as electricity, warranties, support contracts, and eventually replacement.

Public Cloud Benefits:

- Easy and inexpensive to set up (provider has paid for the upfront startup costs)
- Easy to use
- Scalable
- Pay as you go, no wasted resources

Private Cloud benefits:

- Increased control over data, underlying systems, and applications
- Ownership and retention of governance controls
- Assurance of data location, which simplifies legal and compliance requirements

Hybrid Cloud benefits:

- Ability to retain ownership and management of critical tasks and processes
- Reuse technology already owned
- Control critical business components
- Cost-effective by using public cloud for non-critical/non-compliance functions
- Use cloud bursting and disaster recovery functions of the cloud

Abuse of Cloud Services:

- If willing to pay for resources, attackers can use cloud services for harm, such as: Dictionary attacks, DoS attack and Password cracking
- CSPs do watch for some of these activities (specifically DoS) and do work to mitigate them.

Data governance terms to be familiar with:

- Information classification: Description of valuable data categories (confidential, regulated, etc.)
- Information management policies: What activities are allowed for different information classifications (cannot leave premise, cannot be copied to external media, etc.)
- Location and jurisdictional policies: Where can data be geographically located and any regulatory or legal concerns
- Authorizations: Who is permitted to access different types of data
- Custodianship: Who is responsible for managing specific data

Important SLA Components in Cloud:

- Undocumented single points of failure should not exist.
- Migration to another CSP should be permitted within an agreed-upon timeframe.
- If alternate CSPs cannot provide necessary services, an on-premises solution may be required.
- Customer should be able to verify data integrity via automated controls.
- Data backup solutions should allow for granular settings.
- Regular reviews of the SLA should occur to ensure cloud services continue to meet the needs of the business.

Security Considerations for IaaS:

- Controlling network access
- Failover or other redundancy
- Monitoring for availability, security, and audit purposes
- Patching of applications and VM OS

Security Considerations for SaaS:

- Access control to applications
- Controlling devices where application is installed (BYOD)
- Monitoring for availability, security, and audit purposes

Security Considerations for PaaS:

- System and resource isolations (due to multitenancy)
- Access control to applications and permissions
- Secure coding practices for customer build applications
- Monitoring for availability, security, and audit purposes
- Protection against malware

For All of These Cloud Categories:

- Know where your data is
- Review contracts and SLAs so you know what to expect
- Great reference document for keeping up with web application vulnerabilities
- Should read through them and understand prevention methods

Cloud Data Life Cycle Phases:

There are six phases of the Cloud Data Lifecycle:

- Create: Data creation, acquisition or altering. Preferred time to classify data.
- Store: Committing data to storage. At this point, implement security controls to protect data (encryption, access policies, monitoring, logging, and backups).
- Use: Data being viewed or processed not altered. Data is most vulnerable at this point. Controls such as data loss prevention (DLP), information rights management (IRM), and access monitoring should be implemented to protect data at this phase.
- Share: Difficult to manage data once it leaves the organization. DLP and IRM can be helpful to manage what data can be shared.
- Archive: Data no longer actively used is moved to long-term storage. Archived data must still be protected and meet regulatory requirements.
- Destroy: Removal of data from a CSP.

Cloud-Based DLP Considerations:

- Data movement (replication): Can be challenging for DLP systems to deal with
- Administrative access: Discovery and classification can be difficult in dispersed cloud environments
- Performance impact: Network or Gateway DLP solutions can impact network performance, while workstation DLP solutions can slow down endpoints
- CSP approval: May need CSP approval to deploy a DLP solution. If it's a hardware solution, this would be hard to get approval for. If it's CSP product, no approval is necessary. If it's software you're deploying into PaaS, then approval is not likely necessary. If deploying a virtual image DLP into IaaS, it's best to check with the CSP.

Cloud-Specific Risks:

- Management plane breach: Most important risk because this would give the attacker access to the entire infrastructure.
- Resource exhaustion: Oversubscription by the CSP may result in a lack of resources for your cloud services, which may cause an outage
- Isolation control failure: When one tenant is able to access another tenant's resources or affect another tenant's resources.
- Insecure or incomplete data deletion: Be sure to use crypto shredding.
- Control conflict risk: Implementing excessive controls can cause a lack of visibility.
- Software-related risk: Software is prone to vulnerabilities and must be kept up to date.
- Man-in-the-middle attacks: Cloud solutions increase the risk of man-in-the-middle attacks.

Multi-cloud:

MC is where a business wants to spread their IT across multiple clouds, public and private to get the required levels of privacy, security and resilience, avoiding a single vendor lock-in. However, adopting a multi-cloud approach introduces many challenges around the management and interoperability of applications and services.

Benefits of Identity as a Service IDaaS:

• SSO authentication
• Federation
• Granular authorization controls
• Administration
• Integration with internal directory services
• Integration with external services

SSO Technologies:

• Kerberos
• SESAME
• LDAP
• Microsoft Active Directory

OAuth Flow:

• Ask for a request token
• Get Temporary credentials
• Exchange for an access token

Virtualization Risks:

• VM Sprawl
• Sensitive Data within a VM
• Security of Offline and Dormant VMs
• Security of Pre-Configured (Golden Image) VM / Active VMs
• Lack of Visibility Into and Controls Over Virtual Networks
• Resource Exhaustion
• Hypervisor Security
• Unauthorized Access to Hypervisor
• Account or Service Hijacking Through the Self-Service Portal
• The workload of Different Trust Levels Located on the Same Server
• Risk Due to Cloud Service Provider API

Prevent Vulnerabilities in Virtual Machine Infrastructure:

• Make sure that a patch management system is in place.
• Provide the minimum access needed in virtual machines and virtual networks.
• Log and review user and system activities in the virtual environment.
• Pay special attention to how you configure virtual networking devices.
• Consistently capture snapshots or the state of the virtual environment.
• Carefully monitor the number of virtual machines to avoid VM sprawl.
• Protect against VM escape

Prevent Vulnerabilities in Virtual Machine Infrastructure

• Make sure that a patch management system is in place to ensure that all relevant patches are installed. This is especially important for any patches released that apply to the virtualization software itself. Also, carefully determine when and if general operating system patches should also be installed on the host and guests.
• Provide the minimum access needed in virtual machines and virtual networks to meet requirements. This will limit potential damage if security is breached. Monitor access to all environments on a regular basis to prevent unauthorized access.
• Log and review user and system activities in the virtual environment to check for irregular activity and any possible security breaches.
• Pay special attention to how you configure virtual networking devices, enabling network connectivity between systems only when necessary. Note that the security capabilities of virtual networking appliances may not be exactly the same as a physical device. For example, virtual switches in certain modes may fail to isolate traffic between host and guest or guest and guest in a virtual infrastructure.
• Consistently capture snapshots, or the state of the virtual environment at a certain point in time, to provide a quick and easy way to recover the entire environment should it be compromised.
• Carefully monitor the number of virtual machines to avoid VM sprawl, which occurs when the number of virtual machines exceeds the organization's ability to control or manage all of those virtual machines. A compromised VM could easily slip by your notice if you're dealing with VM sprawl. One of the best ways to avoid VM sprawl is to use a VM lifecycle management (VMLM) solution. VMLM solutions provide you with a centralized dashboard for maintaining and monitoring all of the virtual environments in your organization.
• Protect against VM escape, which occurs when an attacker executes code in a VM that allows an application running on the VM to "escape" the virtual environment and interact directly with the hypervisor. The attacker may be able to access the underlying host operating systems and thereby access all other VMs running on that host machine. The best way to protect against VM escape is to ensure that your virtualization software is kept up-to-date. You can also attempt to limit the resource sharing functionality between host and guest.

Provide Secure Session Management:

• Create session IDs only on trusted systems, such as a server.
• In session IDs, do not include any information that is descriptive of the application environment or any user information that would be useful to an attacker performing reconnaissance.
• Make session IDs random and long enough (e.g., 20 bytes or longer) to prevent guesswork or brute force attacks.
• Use the session management controls provided by the server or framework that use algorithms that produce sufficiently random session IDs.
• Set a restrictive domain and path for cookies containing authenticated session IDs.
• Fully terminate the associated session or connection upon logout.
• Provide the ability to log out from all pages protected by authorization.
• Establish a session inactivity timeout that is as short as possible to support business functional requirements.
• Enforce periodic session terminations, even when the session is active, providing warnings to the user as needed.
• Close any sessions established before login and establish a new session after successful login.
• Generate a new session ID and deactivate the old one periodically:
• Upon reauthentication
• If the connection security changes from HTTP to HTTPS
• Locate session IDs only in the HTTP cookie header, and do not expose them in URLs (e.g., GET parameters), error messages, or logs.
• Within an application, consistently utilize HTTPS rather than switching between HTTP and HTTPS.
• Supplement standard session management for sensitive or critical operations such as account management—for example, using per-session strong random tokens or parameters.
• Set the secure attribute for cookies transmitted over a TLS connection.
• Protect session data on the server from unauthorized access by implementing appropriate access controls on the server.
• Apply the HttpOnly attribute to cookies, unless you specifically require client-side scripts within your application to read or set a cookie's value.

Authentication and Authorization Protocols:

- SAML:
  - Authentication and Authorization/Enterprise
  - Single sign-on for enterprise users
- SPML:
  - Account Provisioning/Account Management, SPML paired with SAML
- XACML:
  - Control policies
- OAuth:
  - Resource Access integrated with OpenID
  - API authorization between applications
- OpenID:
  - Authentication and Authorization/Commercial/Mobile App
  - Single sign-on for consumers

MDM solutions include:

- Device enrollment and authentication.
- Remote locks and wipe.
- Locating devices through GPS and other technologies.
- Pushing out OS, app, and firmware updates to devices.
- Preventing root access or jailbreaking of the device.
- Constructing an encrypted container on devices in which to keep sensitive organization data.
- Restricting certain features and services based on access control policies.

Threats in BYOD Environments:

- De-perimeterization
- Unpatched and insecure devices
- Strained infrastructure
- Forensic complications
- Lost or stolen devices

Management Controls for Privacy and Data Protection measures:

- Separation of Duties
- Training
- Authentication and Authorization procedures
- Vulnerability Assessments
- Backup and Recovery processes
- Logging
- Data-retention control
- Secure disposal

Data Protection (How To…):

- Physical Security - Locked doors, security guards, access controls
- Network Security - Authentication, authorization, auditing, firewalls, IDS/IPS
- System Security - Patching, AV, configuration controls, approved applications
- Application Security - Secure coding, code review, design standards
- User Security - Policies, training, provisioning, monitoring, enforcement
- Administrator Security - Policies, supplemental training, provisioning, monitoring, specialized auditing, enforcement

Logging:

Make sure you log security events when you implement application logging. System operators and
security specialists find this information helpful for:

- Detecting attacks and other security-related events
- Obtaining data for incident investigation
- Establishing baselines for security monitoring systems
- Tracking repudiation and implementing related controls
- Monitoring policy violations

Implement Logging:

- Maintain logs on a trusted system, such as a server.
- Protect logs from attackers.
- Log both success and failure of specified security events.
- Ensure log entries that document input data provided by users will not execute as code within the log viewing interface.
- Include information that will be helpful to security analysts:
- Precise time of the event (in UTC format).
- Name or ID of the process that logged the event.
- An informative (if brief) description of the event.
- Name or code for the type of event being logged.
- Do not store sensitive information in logs, including unnecessary system details, session identifiers, or passwords.
- Log the types of events that will be helpful to security analysts:
- Potential security violations, such as file upload virus detection, access of unauthorized ports and protocols, and cryptographic module failures.
- Access to protected resources, including the user, the resource being accessed, and whether the access attempt failed or succeeded.
- Session management failures, such as invalid or expired session tokens.
- Authentication attempts, including the user and whether the attempt failed or succeeded.
- User opt-ins, such as terms of use, consent to use personal data, email lists, and so forth.
- Input validation failures, such as unacceptable length, characters, and encodings.
- Output validation failures, such as invalid data encoding and database record set mismatch.
- Application and related systems startup and shutdown.
- Data file reads, including what portion of data was read, and who read it.
- Data file modifications, including what portion of data was modified, and who modified it.
- Data file deletion, including what portion of data was deleted, and who deleted it.
- Data attribute modification, such as access permissions, labels, ownership, including what data was affected, and who modified it.
- General errors and system events, such as system exceptions, connection and performance issues, errors reported from external services, file system errors, and backend TLS connection failures.
- Performance of any administrative tasks, including changes to settings and configuration, user account management, changes to privileges, enabling or disabling logging or debugging features, viewing user information.
- Network communication, including attempts to use unauthorized ports and protocols.
- Use of any high risk functionality, such as access to payment cardholder data, data import and export, file uploads, and so forth.
- Centralize your logging functions in a secure module that handles logs in a consistent way:
- Uses a cryptographic hash function to validate the integrity of log entries.
- Enables new records to be added, but prevents older records from revision or deletion.
- Uses a standard naming convention for log files, to facilitate sorting and searching.
- Implements functions to automatically verify on a regular basis that logging is still active.
- Restrict unauthorized individuals from accessing logs.
- Synchronize all logging components with a timeserver that has been hardened and isolated from other services.

• Ensure that a mechanism exists to conduct log analysis and that logs are written in a readable format.
• Make secure offsite backups of logs on a regular basis.
• Delete and dispose of log files properly and in accordance with company policy and compliance regulations.

Data Discovery Approaches:

- Big data: A way of analyzing very large data sets to extract information
- Real-time analytics: Looking for patterns of usage
- Agile analytics: Freeform adaptive analysis that focuses on a single problem and doesn't analyze all of the data
- Business intelligence: Analyzing data and presenting useful information to help decision makers

Data Discovery Techniques:

- Metadata: Information about the file (owner, size, create date, etc.)
- Labels: Labels assigned to data by the owner
- Content analysis: Analyzing data content, looking for keywords

Multi-factor Authentication (MFA):

Use multiple factors to authenticate. These factors are based on:

- What they know (password, PIN)
- What they have (token, card, Yubikey)
- What they are (biometrics)

One-time passwords fall under MFA and are highly encouraged for use with first-time logins.

Step-up authentication is also used for MFA when accessing a high-risk transaction or violations have occurred in the transaction:

- Challenge questions
- Out-of-band authentication (SMS, text, phone call, etc.)
- Dynamic knowledge-based authentication (question unique to the individual, previous address, etc.)

Security of Logs:

• Control the volume of data
• Event filtering or clipping level determines the amount of log
• Auditing tools can reduce log size
• Establish procedures in advance
• Train personnel in pertinent log review
• Protect and ensure unauthorized access
• Disable auditing or deleting/clearing logs
• Protect the audit logs from unauthorized changes
• Store/archive audit logs securely

Frameworks:

• Zachman Framework - not specific to security architecture
• Sherwood Applied Business Security Architecture (SABSA) Framework - Chain of traceability
• IT Infrastructure Library (ITIL) - service strategy, service design, service transition, service operations, and continuous service improvement. Processes to allow for IT service management developed by the United Kingdom's Office of Government Commerce
• TOGAF: Model and methodology for the development of enterprise architectures developed by The Open Group
• Six Sigma: Business management strategy that can be used to carry out process improvement
• Capability Maturity Model Integration (CMMI): Organizational development for process improvement developed by Carnegie Mellon

Capability Maturity Model (IRDMO):

• Initial Stage - unpredictable, poorly controlled, and reactive
• Repeatable Stage - characterized for projects, repeatable
• Defined Stage - characterized by the entire organization and is proactive.
• Managed Stage - quantitatively measured and controlled
• Optimizing the Stage - continuous improvement. (Budget)

Capability Maturity Model (IRDMO):

• Level 1: Initial - The software development process is characterized as ad-hoc. Success depends on individual effort and heroics.
• Level 2: Repeatable -Basic project management (PM) processes are established to track performance, cost, and schedule.
• Level 3: Defined - Tailored software engineering and development processes are documented and used across the organization.
• Level 4: Managed - Detailed measures of product and process improvement are quantitatively controlled.
• Level 5: Optimizing - Continuous process improvement is institutionalized.

Other Maturity Models:

• DevOps Maturity Model: Another way to think of an organization's maturity (at least in terms of software development) is to consider how effective it is at integrating its development and operations teams (DevOps). This model is noteworthy in that it focuses on culture and people in addition to development and business issues.

• Open Source Maturity Model (OSMM): For organizations that embrace open-source software, the OSMM allows them to measure and improve the effectiveness of their processes. The focus here is not just on developing (or even just using) open-source software, but on being part of the
movement by developing it, using it, and actively participating in the community.

• Software Product Management Maturity Model: This model focuses on the business issues surrounding the development of software products. For example, it considers issues like market conditions, product lines and portfolios, and partnering agreements.

DevOps:

DevOps and cloud computing work together to help organizations bring new services and applications to market more quickly, at less cost. DevOps is about streamlining the development, while cloud offers on-demand resources, automated provisioning, and easy scaling, to accommodate application changes. Many DevOps tools can be acquired on-demand in the cloud or as part of a larger cloud platform. To support hybrid cloud deployment (workloads with an ability to move between clouds), enterprises should select DevOps platforms with an interface to the cloud providers they will use. DevOps promotes lean and agile delivery of quality software that adds value to business and customers.

DevOps reference:

• Plan and measure
• Develop and test
• Release and deploy
• Monitor and optimize

DevOps Principles:

• Develop and test against production-like systems
• Deploy with repeatable, reliable processes
• Monitor and validate the operational quality
• Amplify feedback loops

DevOps Practices:

• Release planning
• Continuous integration
• Continuous delivery
• Continuous testing
• Continuous monitoring and feedback

Note: DevOps and cloud computing work together to help organizations bring new services and applications to market more quickly, at less cost. DevOps is about streamlining the development, while cloud offers on-demand resources, automated provisioning, and easy scaling, to accommodate application changes. Many DevOps tools can be acquired on-demand in the cloud or as part of a larger cloud platform. To support hybrid cloud deployment (workloads with an ability to move between clouds), enterprises should select DevOps platforms with an interface to the cloud providers they will use.

SOC:

SOC reports most commonly cover the design and effectiveness of controls for a 12- month period of activity with continuous coverage from year to year to meet user requirements from a financial reporting or governance perspective. In some cases, a SOC report may cover a shorter period of time, such as six months. A SOC report may also cover only the design of controls at a specified point in time for a new system/service or for the initial examination (audit) of a system/service.

• SOC1: Focused on Financial Controls
• SOC2: Focused on CIA and Privacy -- Private
• SOC3: Focused on CIA and Privacy – Public

SOC :

• The purpose of a SOC 1 report scope should cover the information systems (both manual and automated) processes that are utilized to deliver the services under review. There are two types of SOC 1 reporting options:

•• SOC 1 Type 1: A design of controls report. This option evaluates and reports on the design of controls put into operation as of a point in time.

•• SOC 1 Type 2: Includes the design and testing of controls to report on the operational effectiveness of controls over a period of time (typically 12 months).

• The purpose of a SOC 2 report is to evaluate an organization's information systems relevant to security, availability, processing integrity, confidentiality, and/or privacy.

•• SOC 2 Type 1: Reports concern policies and procedures that were placed in operation at a specific moment in time.

•• SOC 2 Type 2: Reports concern policies and procedures over a period of at least – systems must be evaluated (normally 6 – 12 months in duration).

This generally makes SOC 2 type 2 reports more comprehensive and useful than type I reports when considering a possible service provider's credentials.

SOC 2 framework includes 5 key sections:

• Security - The system is protected against unauthorized physical and logical access.
• Availability - The system is available for operation and use as committed or agreed.
• Processing Integrity - System processing is complete, accurate, timely, and authorized.
• Confidentiality - Information designated as confidential is protected as committed or agreed.
• Privacy - Personal information is collected, used, retained, disclosed, and destroyed in conformity with the commitments in the entity's privacy notice.

Information Security Strategies:

• Strategic planning – Long-term (3 to 5 years) and must be aligned with business objectives.
• Tactical planning – Short-term (6 to 18 months) used to achieve specific goals. May consist of multiple projects.
• Operational and project planning – Specific plans with milestones, dates, and accountabilities provide communication and direction for project completion.

Confinement, Bounds, and Isolation:

• Confinement- restricts a process to reading from and writing to certain memory locations.
• Bounds - are the limits of memory a process cannot exceed when reading or writing.
• Isolation - is the mode a process runs in when it is confined through the use of memory bounds.

Markup Language:
• GML: Generalized Markup Language - a Top level markup language
• SGML: Standardized Generalized Markup Language - Derived from GML
• SPML: Service Provisioning Markup Language -Allows exchange of provisioning data between systems. SPML: XML based format for exchanging user and resource information and controlling provisioning.
• SAML: Security Assertion Markup Language - Standard that allows the exchange of Authentication and Authorization data to be shared between security domains. SAML can expose the system to poor identification or authorization. SAML: provides an XML-based framework for exchanging security-related information over networks.
• XACML: Extensible Access Control Markup Language - Used to express security policies and access rights provided through web services and applications
• XML: Can include tags to describe data as anything desired. Databases from multiple vendors can import and export data to and from an XML format, making XML a common language used to exchange information. XML is vulnerable to injection attacks. XML is a universal format for storing information.

Life Cycle of Evidence:
• Collection and Identification
• Storage, preservation, and transportation
• Presentation in court
• Return of the evidence

Equipment Life Cycle:
• Defining requirements
• Acquiring and implementing
• Operations and maintenance
• Disposal and decommission

Decommissioning:
When an organization decides to decommission a system or service or when they reach the end of their service life, these services must be decommissioned without leaving data, other systems, or personnel at risk. Systems and services must be properly terminated to eliminate the risk to remaining systems. There are some steps in the process of decommissioning with conversion outlined below:
• Migration Plan
• Perform Migration
• Decommissioning Plan
• Perform Decommissioning
• Post Decommissioning Review

Data Removal:
• Erasing: delete operation
• Clearing: overwriting operation
• Purging: more intensive form of clearing by repetition
• Declassification: purge media to be suitable for use for the secure environment
• Sanitization: combination of a process that removes data from a system or media
• Degaussing: use of a strong magnetic field
• Destruction: crushing, Incineration, Shredding, disintegration

Data Archiving:
• Format
• Regulatory requirements
• Testing

Establish Information and Asset Handling Requirements:
• Secure disposal of media: Media containing sensitive data has to be disposed off in a secure manner. Shredding in the case of paper documents and pulverizing in the case of digital media are some of the methods used in media disposal.
• Labeling: Appropriate labeling is important for sensitive data without disclosing the type of
content.
• Access Restrictions: Understand the principle to adopt in designing and implementing access restrictions to sensitive data.
• Authorized Recipient's Data: Recipients who are authorized to access the data should be documented and approved.
• Storage of media: Media storage should be accordingly manufacturers' specifications and industry best practices.
• Data Distribution: Appropriate controls should be established to ensure that the data is distributed only to approved and authorized personnel with respect to the authorized recipient's list.
• Clear Marking Marking on sensitive data has to be clear and understandable for appropriate
identification and handling. Marking may use codes to compare labeling that may only
be used for identification purposes.
• Review of Distribution Lists: Periodic review of the distribution lists is necessary to ensure that the data is shared only with authorized individuals.
• Publicly Available Sources: Suitable controls should be proven to ensure that sensitive data is not disclosed or posted to publicly available repositories or websites.

Media control:
• Accurately and promptly mark all data storage media
• Ensure proper environmental storage of the media
• Ensure the safe and clean handling of the media
• Log data media to provide a physical inventory control

Steps Data retention:
• Evaluate Statutory Requirements, Litigation obligations, and business needs
• Classify types of records
• Determine retention periods and destruction policies
• Draft and justify record retention policy
• Train staff
• Audit retention and destruction practices
• Periodically review policy
• Document policy, implementation, training, and audits

Retention policies should address:
• Storage
• Retention
• Destruction / Disposal

Documentation:
All kind of documentation should be subject to an effective version control process as well as a standard approach to marking and handling; and conspicuously labeled with classification level, revision date and number, effective dates, and document owner.

General Data Backup Considerations:
• The scope of Backups/ Total size
• Importance
• Security
• Frequency of change
• Recovery time
• Testing the Integrity of Backups

Sensitivity vs. Criticality:
• Sensitivity describes the amount of damage that would be done should the information be disclosed
• Criticality describes the time sensitivity of the data. This is usually driven by the understanding of how much revenue a specific asset generates, and without that asset, there will be lost revenue

Factors effective Biometrics Access Control System:
• Accuracy
• Speed/Throughput
• Data storage requirements
• Reliability
• Acceptability

Downsides biometric:
• User acceptance
• Enrollment timeframe
• Throughput
• Accuracy over time

Defense-in-depth strategy:
• Developing security policies, procedures
• Addressing security throughout the lifecycle
• Implementing a network topology has multiple layers
• Providing logical separation between the corporate and network devices
• Employing a DMZ network architecture
• Ensuring that critical components are redundant and are on redundant networks.
• Designing critical systems for graceful degradation (fault tolerant)
• Disabling unused ports and services
• Restricting physical access to network and devices.
• Restricting user privileges
• Considering the use of separate authentication mechanisms and credentials
• Using modern technology
• Implementing security controls
• Applying security techniques
• Expeditiously deploying security patches
• Tracking and monitoring audit trails

Physical Security:
• Protecting life is the primary goal of physical security
• Physical security helps prevent operational interruptions
• The primary goal of the physical program is facility access control
• Arrange barriers in layers with progressive security closer to center/highest protective area
• Conduct a security risk/vulnerability assessment to identify threats (natural and man-made) to assets and impacts of the loss
• During assessment address security control during/after hours, access control, surveillance, policies/procedures, BCP, etc.
• Apply defense in depth

System engineering management:
• Decision Analysis
• Technical Planning
• Assessment Requirements
• Configuration, Interface
• Technical Data
• Risk Management

Industrial control system key-components (ICS):
• Control Loop
• Human-Machine Interface (HMI)
• Remote Diagnostics and Maintenance Utilities

Major control components of industrial control systems (ICS):
• Control Server
• SCADA Server or Master Terminal Unit (MTU)
• Remote Terminal Unit (RTU)
• Programmable Logic Controller (PLC)
• Intelligent Electronic Devices (IED)
• Human-Machine Interface (HMI)
• Data Historian
• Input / Output (IO) Server

Platform vulnerabilities in industrial control systems (ICS):
• Platform Configuration Vulnerabilities
• Platform Hardware Vulnerabilities
• Platform Software Vulnerabilities
• Platform Malware Protection Vulnerabilities

Developing a Comprehensive Security Program for (ICS):
• Obtain senior management buy-in
• Build and train a cross-functional team
• Define charter and scope
• Define specific ICS policies and procedures
• Define and inventory ICS assets
• Perform a risk and vulnerability assessment
• Define the mitigation controls
• Provide training and raise security awareness for ICS staff

ICS Security:
• Disable unnecessary ports & services
• Network Segmentation
• Enforce Encryption where applicable
• Enforce patch management
• Risk management application to ICS
• Implementation of least privileges policy
• Audits
• Redundancy & Fault Tolerance

Big Data:
Data collections that are so large and complex that they are difficult for traditional database tools to manage. Businesses are often prompted to restructure their existing architecture to handle it.

Big Data:
Cloud Secure Alliance (CSA) has categorized the different security and privacy challenges into four different aspects of the Big Data ecosystem. These aspects are Infrastructure Security, Data Privacy, Data Management and, Integrity and Reactive Security. Each of these aspects faces the following security challenges, according to CSA:
• Infrastructure Security
  •• Secure Distributed Processing of Data
  •• Security Best Actions for Non-Relational DataBases
• Data Privacy
  •• Data Analysis through Data Mining Preserving Data Privacy
  •• Cryptographic Solutions for Data Security
  •• Granular Access Control
• Data Management and Integrity
  •• Secure Data Storage and Transaction Logs
  •• Granular Audits
  •• Data Provenance
• Reactive Security
  •• End-to-End Filtering & Validation
  •• Supervising the Security Level in Real-Time

Common threats to Big Data:

- Breach of privacy
- Privilege escalation
- Repudiation
- Forensic complications

Secure life cycle for big data

The life cycle of big data has six main stages: creation and discovery, access and data flow, process, share, store, and destroy.

- The key challenges in creation and discovery are:
  - Identifying all endpoints in the network
  - Identifying intellectual property and determining the value and business impact of each data in the big data cluster.
  - Defining data provenance.
- The security challenges in access and data flow are:
  - Implementing security in distributed frameworks.
  - Implementing granular access controls
  - Defining security controls for non-relational data sources.
  - Identifying end-to-end data flow.
- Security challenges while data processing are:
  - Implementing scalable, privacy and security during data mining and data analytics.
  - Implementing granular data audits.
- The security challenges while sharing data are:
  - Implementing granular data audits.
  - Implementing reactive security to secure the integrity
- The security challenges while storage data are:
  - Implementing secure data storage and transaction data logs and files.
- Data disposal is the most crucial stage in the life cycle of big data. Data in the wrong hands may be catastrophic. Organization-level security policies to implement secure data disposal methods and removal of access rights on employee/user exit interviews should be in place to ensure the data is available only to authorized users.

Challenges of current Big Data:

- There are limited levels of protection in the majority of distributed systems computations.
- Security solutions are not being able to tackle the demand with several non-relational databases constantly
- There is a lack of appropriate security processes for the transfer of automated data.
- System updates, audits, patches are not always carried out.
- Information coming in should be constantly validated, to ensure its credibility and accuracy
- The attack on systems that contain sensitive information of the customers can put the customers at risk.
- Some organizations do not deploy any kind of access controls to differentiate between the confidentiality
- Monitoring and tracking of systems are difficult with the current scale of Big Data application.

Machine learning (ML), Blockchain and artificial intelligence (AI):

Using pattern recognition and computational learning to make predictions. Many cloud vendors are now offering ML and AI as a service. Cloud vendors have the resources to build out environments suited for this type of data analysis.
Blockchain: A protocol that uses a decentralized framework to maintain integrity within the data Cloud was originally the offloading of service from on-premises to a cloud vendor's premises where customers use resources from one or more data centers. Blockchain could be used to manage globally distributed workloads between data centers so that the data resides in multiple data centers at once. Not only would this allow for a new type of decentralized cloud, but it would also be used to guarantee integrity of the data.

Quantum computing:

Quantum computing gets its massive compute power by tapping into quantum physics and not the use of micro-transistors. Traditional computing uses the values of 0 and 1 in bits, but quantum computing can store multiple values in qubits. Vendors such as Rigetti, Google, IBM, and Microsoft have made quantum CPUs, but they are still in the infancy of the projects and are working to build applications that can take advantage of the processing power so they can measure the processing power. Eventually, cloud service providers will offer quantum computing services to their customers. We can only imagine that with the shared resource framework, providers will be able to offer quantum computing services at a much more affordable rate than attempting to purchase a quantum computing server.

Artificial Intelligence, Machine Learning and Deep Learning:

- Artificial intelligence: Any technique which enables computers to mimic human behavior
- Machine Learning: Subset of AI techniques which use statistical methods to enable machines to improve with experiences.
- Deep Learning: Subset of ML, which make the computation of multi-layer neural networks feasible.

Artificial Intelligence (AI):

- Expert Systems
- Artificial Neural Networks
- Real Neural Networks
- Bayesian Filtering
- Genetic Algorithms and Programming

AI/ML projects:

They are heavily based on data, and often that data involves personally identifiable information (PII), which must be protected to ensure the privacy of the people described by that data. PII is associated with an individual person, such as an employee, customer, or patient. PII can be used to uniquely identify, contact, or locate an individual. Examples include a person's name, email address, home address, Social Security number (even if it's just the last 4 digits), and so forth.

OWASP Top 10 IoT Vulnerabilities:
- Insecure Web Interface
- Insufficient Authentication/Authorization
- Insecure Network Services
- Lack of Transport Encryption/Integrity Verification
- Privacy Concerns
- Insecure Cloud Interface
- Insecure Mobile Interface
- Insufficient Security Configurability
- Insecure Software/Firmware
- Poor Physical Security

Following is OWASP's list of the Top 10 Privacy Risks:
- P1: Web Application Vulnerabilities
- P2: Operator-sided Data Leakage
- P3: Insufficient Data Breach Response
- P4: Insufficient Deletion of Personal Data
- P5: Non-transparent Policies, Terms and Conditions
- P6: Collection of data not required for the primary purpose
- P7: Sharing of data with third party
- P8: Outdated personal data
- P9: Missing or Insufficient Session Expiration
- P10: Insecure Data Transfer

OWASP:
The Open Web Application Security Project (OWASP) has incorporated these principles into its list of ten "Security by Design Principles" The principles are:
- Minimize attack surface area.
- Establish secure defaults.
- Least privilege.
- Defense in depth.
- Fail securely.
- Don't trust services.
- Separation of duties.
- Avoid Security by Obscurity.
- Keep security simple.
- Fix security issues correctly

OWASP threat risk modeling process steps:
- Identify Security Objectives
- Survey the Application
- Decompose it
- Identify Threats
- Identify Vulnerabilities

Provide Secure Authentication and Session Management
When authentication and session management are implemented incorrectly, an attacker can compromise passwords, keys, or session tokens, or exploit other implementation flaws to assume the identity of another user. Resources used to establish and maintain secure sessions (such as Session IDs, passwords, and other credentials) must be properly protected. If attackers obtain control of these resources, they can gain privileged access, like an authorized user. To avoid authentication and session management defects, make sure your software:
- Bases its authentication and session management capabilities upon a single set of strong authentication and session management controls. Instead of writing your own routines to authenticate, create and end sessions, store tokens, and so forth, consider using well-tested libraries and frameworks such as the ESAPI Authenticator and User APIs provided by OWASP.
- Requires strong passwords. More complex passwords are harder to identify through brute force or automated methods. For example, you might require a minimum length and the use of alphabetic, numeric, and special symbols in a user's password.
- Requires users to change passwords periodically and not reuse old passwords.
- Does not log passwords entered on failed login attempts. Since a legitimate user may occasionally mistype their password, keeping a log of "almost correct" passwords can provide clues to an attacker who manages to gain access to a compromised security log.
- Blocks repeated failed attempts. Brute force attacks are facilitated when different passwords can be tried repeatedly and quickly. Disable the account (at least temporarily) after a few failed logins, log a security event, and notify system operators that an attack may be underway. Help users monitor their own account security. For example, when users successfully log in, show them the date and time they last logged in, as well as the number of failed access attempts on their account since the last login.
- Provides a single, careful mechanism through which passwords can be changed. Require users to re-authenticate (using their current password) when changing their password (or any other account information), even if they are already logged in.
- Does not store passwords. If you must validate a password, store a hash, not the password itself.

Provide a Secure Web Interface
To avoid this defect, make sure web-based administrative consoles:
- Are configured to install with the safest default settings, assuming that many users will not change the configuration.
- Enable default user names and passwords to be changed, and prompt the user to do so upon first use.
- Require strong passwords.
- Provide an account lockout feature after a certain number of failed access attempts.
- Do not include common web vulnerabilities (XSS, CSRF, SQL injection, and so forth).
- Use HTTPS to protect transmitted information.
- Use web application firewalls.
- Provide a means to receive upgrades and security fixes.
- Adhere to all general patterns for preventing web vulnerabilities.

Provide Secure Authentication
- Require authentication for all pages and resources that are not meant to be public.
- Reauthenticate users prior to performing critical operations.
- Authenticate all connections to external systems involving sensitive information or functions.
- Enforce all authentication controls on a trusted system, such as a server.
- Use standard, tested authentication services whenever possible.
- If using third-party code for authentication, inspect the code carefully to ensure it does not contain malicious code.
- Use a centralized implementation for all of your authentication controls.
- Keep authentication logic separate from the resource being requested.
- Use redirection to and from the centralized authentication controls.
- Ensure that authentication controls fail to the most secure state.
- Ensure that administrative and account management functions are at least as secure as the primary authentication mechanism.
- Use strong hashing algorithms for credential stores.
- Ensure that the credential store is writeable only by the application.
- Validate authentication data only after all inputs are provided, especially for sequential authentication implementations.
- When an authentication fails, give no clues as to which part of the authentication data was incorrect. For example, show no

differences between the message displayed for incorrect user name and incorrect password.
• Encrypt and store authentication credentials that your software uses to access external services in a protected location on a trusted system—not within the source code.
• Use only secure channels to transmit authentication credentials—for example, use a POST request over HTTPS.
• Disable "remember me" functionality for password fields.
• Report the date and time of the last successful or unsuccessful login attempt of a user account to the user at the next successful login.
• Monitor the system for attacks against multiple user accounts that use the same password.
• Use multifactor authentication for highly sensitive or high value transactional accounts.
• Do not permit concurrent logins using the same ID.

Physically Secure IoT Devices:
IoT devices should be protected from direct physical access by an attacker. Make sure your devices:
• Provide external ports (e.g., USB) only when absolutely essential.
• Limits access to external ports (through an authentication process, for example).
• Have operating systems that are properly protected.
• Can be configured to limit administrative capabilities, preferably defaulting to least privilege.
• Are tamper resistant.
• Do not expose any testing or debugging interfaces that can be used to gain unauthorized access.
• Account for the transfer of ownership of devices to ensure that data is not transferred along with the ownership.

IoT architecture:
• The perception layer
• The network layer
• The application layer

IoT Characteristics:
• Existence
• Sense of self
• Connectivity
• Interactivity
• Dynamicity
• Scalability
• Limitations of Computational
• Limitations of Resources

The IoT building block consists of five main modules:
• Sensor Module
• Processing Module
• Actuation Module
• Communication Module
• Energy Module

IoT Attack Areas:
The following are the most common attack areas for IoT network:
• Access Control.
• Firmware Extraction.
• Privileges Escalation.
• Resetting to an insecure state.
• Web Attacks.
• Firmware Attacks.
• Network Services Attacks.
• Unencrypted Local Data Storage.
• Confidentiality and Integrity issues.
• Cloud Computing Attacks.
• Malicious updates.
• Insecure APIs.
• Mobile Application threats.

Securing IoT: (Seven Steps to Minimize IoT Risk in the Cloud)
• Secure Cloud Infrastructure
• Leverage Standards-Based Best Practices
• Design for Security
• Secure IoT Devices
• Secure Device Connections
• Secure IoT Services and Apps
• Secure Users and Access

IoT Device Security Challenges:
• IoT products may be deployed in insecure or physically exposed environments
• Security is new to many manufacturers and there is limited security planning in development methodologies
• Security is not a business driver and there are limited security sponsorship and management support in the development of IoT products
• There is a lack of defined standards and reference architecture for secure IoT development
• There are difficulties recruiting and retaining requisite security skills for IoT development teams, including architects, secure software engineers, hardware security engineers, and security testing staff
• The low price point increases the potential adversary pool
• Resource constraints in embedded systems limit security options

Guidance for Secure IoT Development:
• Secure Development Methodology
• Secure Development and Integration Environment
• Identity Framework and Platform Security Features
• Establish Privacy Protections
• Hardware Security Engineering
• Protect Data
• Secure Associated Apps
• Protect Interfaces/APIs
• Provide Secure Update Capability
• Implement Secure Authn/z
• Establish Secure Key Management
• Provide Logging Mechanisms
• Perform Security Reviews

IoT Security (BEST PRACTICES):
- Make hardware tamper resistant
- Provide for firmware updates/patches
- Perform dynamic testing
- Specify procedures to protect data on device disposal
- Use strong authentication
- Use strong encryption and secure protocols
- Minimize device bandwidth
- Divide networks into segments
- Protect sensitive information
- Encourage ethical hacking and vulnerability disclosure
- Institute an IoT Security and Privacy Certification Board

Product vendors/developers should consider steps below to improve IoT security:
- Secure web/desktop/mobile applications with proper authentication and authorization.
- If feasible, Implement and enable 2-factor authentications by default, it will considerably improve IoT device security.
- Follow secure coding methods and always perform input validation to avoid Cross-site scripting (XSS), SQL injection and Buffer Overflow vulnerabilities.
- Enforce an effective password policy
- Use captcha, account lockout policy methods to avoid brute force attacks.
- Vendors should provide security updates including details on security fixes, the impact of the vulnerability and provide simple steps to deploy security updates.
- If feasible, always use encryption for communication.
- Ensure regular backups (at least two or more data) in a secure place.
- Avoid information disclosure. i.e avoid publishing customer's data
- While adding new features, vendors should make sure it will not create security hole.
- Vendors should think on ease of use vs. security
- Apply OWASP Top 10 IoT Vulnerabilities should be addressed, during IoT design.

Types of tests that can be employed for IoT device developments:
- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Interactive Application Security Testing (IAST)
- Attack Surface and Vectors
- 3rd Party Library
- Fuzzing
- Customized per threat vector

IoT Forensics Challenges:
- The Investigation Framework
- Diversity of Devices
- IoT Constraints
- Lack of Standardization
- Improper Evidence Handling
  - Evidence identification, collection, and preservation
  - Evidence analysis and correlation
- Securing the Chain of Custody

Attacks in IoT:
- Node Tampering / Node Compromised
- Denial of Service (DoS)
- Distributed DoS
- Device Spoofing
- The Breach of Privacy
- Malware
- Application-based Attacks
- Man in the Middle Attacks

NIST:
- NIST 800-12 NIST Handbook Intro to Computer Security
- NIST 800-13 Telecomm Security Guidelines for Telecomm
- NIST 800-14 Generally Accepted Principles and Practices
- NIST 800-18 AUP / Rules of Behavior
- NIST 800-30 Risk Management/Assessments
- NIST 800-34 Contingency Planning
- NIST 800-37 Risk Management Framework
- NIST 800-40 Creating a Patch and Vulnerability Management
- NIST 800-41 Guidelines on Firewalls and Firewall Policy
- NIST 800-44 Guidelines on Securing Public Web Servers
- NIST 800-45 Guidelines on Electronic Mail Security
- NIST 800-47 Security Guide for Interconnecting IT Systems
- NIST 800-48 Guide to Securing Legacy IEEE 802.11 Wireless
- NIST 800-50 Building an IT Security Awareness
- NIST 800-53 Security and Privacy Controls for Federal IS
- NIST 800-54 Border Gateway Protocol Security
- NIST 800-55 Security metrics IS
- NIST 800-57 Recommendation for Key Management
- NIST 800-60 Guide for Mapping Types of Information
- NIST 800-61 Computer Security Incident Handling
- NIST 800-63 Electronic Authentication
- NIST 800-64 Security Considerations in SDLC
- NIST 800-66 Healthcare privacy issues
- NIST 800-86 Guide to Integrating Forensic Techn. into IR
- NIST 800-82 Guide to Industrial Control Systems (ICS)
- NIST 800-83 Guide to Malware Incident Prevent and Handling
- NIST 800-86 Guide to Integrating Forensic Tech. into IR
- NIST 800-88 Media Sanitization
- NIST 800-94 IDS/1PS
- NIST 800-115 IS Security Testing and Assessment
- NIST 800-119 Guidelines for Secure Deployment of IPv6
- NIST 800-122 Protect PII
- NIST 800-137 Information Security Continuous Monitoring
- NIST 800-145 Cloud computing

ISO:
- ISO 7498: OSI Model
- ISO 27000: ISMS-Overview and Vocabulary
- ISO 27001: ISMS-Requirement
- ISO 27002: Code of practice
- ISO 27003: ISMS implementation
- ISO 27004: Measurement and metrics framework
- ISO 27005: Risk management
- ISO 27006: Certification body requirements
- ISO 27007: ISMS-Auditing
- ISO 27008: Information Security Control
- ISO 27011: ISMS guideline telecom organization
- ISO 27014: Governance of information security
- ISO 27017: Use of cloud services
- ISO 27018: Cloud privacy protection overview
- ISO 27031: Communications technology readiness for BC
- ISO 27032: Cyber Security Resilience
- ISO 27034: Security applications
- ISO 27035: Security incident management

• ISO 27037: Covers identifying, gathering, and preserving DE
• ISO 27799: Directives on protecting personal health information
• ISO 31000: Risk Management Framework
• ISO 22301: BCM - Business continuity
• ISO 15408: Common Criteria
• ISO 28000: Supply Chain Management
• ISO 42010: Systems and Software Engineering Architecture
• ISO 14443: Smart card standardizations

IEEE:
• IEEE 802.11: Wireless LANs
• IEEE 802.15: Wireless PANs
• IEEE 802.16: Broadband Wireless MANs
• IEEE 802.20: Mobile Broadband Wireless Access

Wireless:
IEEE proposed the 802.11 standards for wireless communications. Various versions have been developed in wireless networking hardware, including 802.11a 802.11b, 802.11g, 802.11n, 802.11ac as described in the table below:
802.11 2 Mbps 2.4 GHz
802.11a 54 Mbps 5 GHz
802.11b 11 Mbps 2.4 GHz
802.11g 54 Mbps 2.4 GHz
802.11n 200+ Mbps 2.4 GHz or 5 GHz
802.11ac 1 Gbps 5 GHz

ISO 27002 includes:
• Security Policy
• Organization and Information Security
• Asset Management
• Human Resources Security
• Physical and Environmental Security
• Communications and Operations Management
• Access Control
• Information Systems Acquisitions, Development, and Maintenance
• Information Security Incident Management
• Business Continuity Management
• Compliance

Important FISMA Features:
• Periodic risk assessments.
• Policies and procedures based on assessments.
• Qualitative risk rating-data-driven security model.
• Subordinate plans for information security for networks, facilities, and other subsystems.
• Security awareness training for personnel.
• Periodic testing and evaluation of the effectiveness of information security policies, procedures, practices, and controls at least annually.
• A process to address deficiencies in information security policies (POAM).
• Procedures for detecting, reporting, and responding to security incidents.
• Procedures and plans to ensure continuity of operations for information systems that support the organization's operations and assets.

Important HIPAA Features:
• Electronic Transaction and Code Sets Standards: Requires the same health care transactions, code sets, and identifiers.
• Privacy Rule: Provides federal protections for personal health information held by covered entities and gives patients an array of rights with respect to that information.
• Security Rule: Specifies administrative, physical, and technical safeguards for covered entities to use to assure the confidentiality, integrity and availability of electronic protected health information.
• National Identifier Requirements: Requires that health care providers, health plans, and employers have standard national numbers that identify them on standard transactions.
• Enforcement Rule: Provides standards for enforcing all the Administrative Simplification Rules.

Important HITECH Features:
• Expansion of HIPAA security standards to "business associates" that perform activities involving the use or disclosure of individually identifiable health information.
• Increased civil penalties for "willful neglect."
• Data-breach notification requirements for unauthorized uses and disclosures of "unsecured PHI."
• Stronger individual rights to access electronic medical records and restrict the disclosure of certain information.
• New limitations on the sale of protected health information, as well as marketing and fundraising communications.

EU Data Protection Directive Features:
• Notice: Data subjects should be given notice when their data are being collected.
• Purpose: Data should only be used for the purpose stated.
• Consent: Data should not be disclosed without the subject's consent.
• Security: Collected data should be kept secure from any potential abuses.
• Disclosure: Data subjects should be informed as to who is collecting their data.
• Access: Data subjects should be allowed to access their data and make corrections to any inaccurate data.
• Accountability: Data subjects should have an available method to hold data collectors accountable for following these six principles above.

COBIT :
• Principle 1: Meeting Stakeholder Needs
• Principle 2: Covering the Enterprise End to End
• Principle 3: Applying a Single, Integrated Framework
• Principle 4: Enabling a Holistic Approach
• Principle 5: Separating Governance from Management

ITIL Benefits:
• Increased user and customer satisfaction with IT services.
• Improved service availability, directly leading to increased business profits and revenue.
• Financial savings from reduced rework, lost time, improved resource management and usage.
• Improved time to market for new products and services.
• Improved decision-making and optimized risk.

OECD:
Organization for Economic Cooperation and Development (OECD) suggests that privacy laws include:
• Collection limitation principle
• Data quality principle
• Purpose specification principle
• Use limitation principle
• Security safeguards principle
• The openness principle

Information Security:
The key goal of information security is to reduce adverse impacts on the organization to an acceptable level. Following are some other security management framework & methodologies for security professionals, which includes development standards, security architect, security controls, governance methods & management process:
•ISO/IEC 17799:2005 Information technology - Security techniques - Code of practice for information security management
• ISO/IEC 27000 Series family of Information Security Management Systems
• ISO/IEC 27001 Information Security Management
• ISO/IEC 27002 Code of practice for information security controls
• Common Criteria (CC) or ISO/IEC 15408
• Information Technology Infrastructure Library (ITIL)
• Zachman framework
• TOGAF
• DoDAF
• MODAF
• COBIT

The structure of the TOGAF documentation:
• PART I: (Introduction)- This part provides a high-level introduction to the key concepts of enterprise architecture and,in particular, the TOGAF approach. It contains the definitions of terms used throughout TOGAF and release notes detailing the changes between this version and the previous version of TOGAF.
• PART II: (Architecture Development Method) - This part is the core of TOGAF. It describes the TOGAF Architecture Development Method (ADM), a step-by-step approach to developing an enterprise architecture.
• PART III: (ADM Guidelines and Techniques) - This part contains a collection of guidelines and techniques available for use in applying TOGAF and the TOGAF ADM.
• PART IV: (Architecture Contenf Framework) - This part describes the TOGAF content framework, including a structured meta-model for architectural artifacts,. The use of re-usable architecture, building blocks, and an overview of typical architecture deliverables.
• PART V: (Enterprise Continuum & Tools) - This part discusses appropriate taxonomies and tools to, categorize and store the outputs of architecture activity within an enterprise.
• PART VI: (TOGAF Reference Models) - This part provides a selection of architectural reference models, which includes the TOGAF Foundation Architecture, and the Integrated Information Infrastructure Reference Model (III-RM).
• PART VII: (Architecture Capability Framework) - This part discusses the organization, processes, skills, roles, and responsibilities required to establish and operate an architecture function within an enterprise.

SABSA:
SABSA is comprised of a series of integrated frameworks, models, methods, and processes used independently, or as a holistic integrated enterprise solution, including:
• Business Requirements Engineering Framework (known as Attributes Profiling)
• Risk and Opportunity Management Framework
• Policy Architecture Framework
• Security Services-Oriented Architecture Framework
• Governance Framework
• Security Domain Framework
• Through-life Security Service Management & Performance Management Framework

The GDPR defines three relevant entities:
• Data subject The individual to whom the data pertains
• Data controller Any organization that collects data on EU residents
• Data processor Any organization that processes data for a data controller

GDPR:
GDPR requires that organizations adhere to the rules of Privacy by Design. Privacy by Design is an approach to software development that takes privacy into account throughout every phase of development. The underlying premise of Privacy by Design is not simply protecting data, but as much as possible designing systems so data doesn't need protection—for example, minimizing data collected in the first place.

Identify Personal Data as Defined by GDPR:
GDPR defines personal data as "any information relating to an identified or identifiable natural person (data subject). An identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier" such as:
• A name
• An identification number
• Location data
• An online identifier
• One or more factors specific to the data subject's physical, physiological, genetic, mental, economic, cultural, or social identity

The GDPR set of protected types of privacy data:

- Name
- Address
- ID numbers
- Web data (location, IP address, cookies)
- Health and genetic data
- Biometric data
- Racial or ethnic data
- Political opinions
- Sexual orientation

Key provisions of the GDPR include:

- Consent Data controllers and data processors cannot use personal data without explicit consent of the data subjects.
- Right to be informed Data controllers and data processors must inform data subjects about how their data is, will, or could be used.
- Right to restrict processing Data subjects can agree to have their data stored by a collector but disallow it to be processed.
- Right to be forgotten Data subjects can request that their personal data be permanently deleted.
- Data breaches Data controllers must report a data breach within 72 hours of becoming aware of it.

CIS:

The Center for Internet Security's (CIS) Critical Security Controls for Effective Cyber
Defense (CSC) is a recommended set of actions for cyber defense that provide specific and actionable ways to stop today's most pervasive and dangerous attacks.

- Basic CIS Controls
  - Inventory and Control of Hardware Assets
  - Inventory and Control of Software Assets
  - Continuous Vulnerability Management
  - Controlled Use of Administrative Privileges
  - Secure Configuration for Hardware and Software for Mobile Devices, Laptops, Workstations and Servers
  - Maintenance, Monitoring and Analysis of Audit Logs
- Foundational CIS Controls
  - Email and Web Browser Protections
  - Malware Defenses
  - Limitation and Control of Ports, Protocols and Services
  - Data Recovery Capabilities
  - Secure Configuration for Network Devices, such as Firewalls, Routers and Switches
  - Boundary Defense
  - Data Protection
  - Controlled Access Based on the Need to Know
  - Wireless Access Control
  - Account Monitoring and Control
- Organizational CIS Controls
  - Implement a Security Awareness and Training Program
  - Application Software Security
  - Incident Response and Management
  - Penetration Tests and Red Team Exercises

Meet PCI DSS Requirements:

- Maintain secure networks. Enable payment card transactions to be conducted safely over the
network. Employ user-friendly firewalls to protect cardholder information without inconveniencing cardholders. Change all passwords and PIN codes for hosts and network
devices from default values to values that cannot be guessed by an attacker.
- Secure cardholder data in transit and in storage. Encrypt sensitive cardholder data such as
social security numbers, birth dates, phone numbers, addresses, and so forth.
- Provide malware protection on client and host systems. Keep malware protection up to date.
Regularly update and patch operating systems and other software dependencies.
- Restrict access to cardholder data to authorized personnel. All users should have a unique ID on the system, so their activities can be logged. Provide physical protection of data in systems and in hard copy. Use document shredders and provide locks on dumpsters to prevent unauthorized access.
- Continually monitor for vulnerabilities. This includes all networks, systems, and applications.
Regularly scan memory and storage to detect potential threats.
- Implement and follow a comprehensive information security policy. Implement systems to ensure that policies are understood and followed by everyone. Impose penalties for noncompliance. Perform auditing on a regular basis.

Provide Sufficient Attack Protection:

The majority of applications and APIs lack the basic ability to detect, prevent, and respond to both manual and automated attacks. Attack protection goes far beyond basic input validation and involves automatically detecting, logging, responding to, and even blocking exploit attempts. Application owners also need to be able to deploy patches quickly to protect against attacks. To avoid this defect, make sure your software:

- Detects attacks and responds appropriately. For example, events might occur that a legitimate user is not likely to cause (such as high-speed input, odd input patterns, repeated requests, etc.). The application can help to protect itself, data, and users by monitoring for such events and providing appropriate interventions, such as ignoring requests, blocking IP addresses, or IP ranges, logging and notifying the user and/or system operator, disabling user accounts, and so forth.
- Is patched quickly. Push protections out quickly. When you can't push out patches and updates immediately, implement provisional protections such as blocking certain traffic patterns to prevent vulnerabilities from being exploited.

Protect Privacy:

To ensure that personal information and privacy are protected:

- Minimize data collection.
- Consult with data scientists and legal and compliance teams to determine risk of data collection and storage.
- Provide end users the option to specify what data will be collected.
- Anonymize collected data.
- Use encryption to protect all collected personal data at rest and in transit.
- Ensure that collected personal information is accessible only by authorized users.
- Ensure that a data retention policy is in place.

Protect Sensitive Information:
Some information may require special care and handling in your application to protect users. Identify any information that is sensitive, and apply appropriate controls to ensure it remains private. A good place to start is to always consider all personally identifiable information (PII) sensitive, as it can be used to establish a person's identity and might be used to cause them substantial harm, embarrassment, inconvenience, or unfairness. Refer to privacy guidelines for your country, municipality, or organization for specific lists of PII you may be legally required to protect. A typical list is provided here.
• User name
• Email address
• Home address
• Phone number
• Social Security number (even if it's just the last 4 digits)
• Driver's license or state ID#
• Passport number
• Alien registration number
• Financial account number
• Biometric identifiers
• Citizenship or immigration status
• Medical information
• Ethnic or religious association
• Sexual orientation
• Account passwords
• Date of birth
• Criminal history
• Mother's maiden name

Anonymize Personal Data:
To anonymize personal data:
• Use one of the following techniques to mask the identifying data:
• Replacement—Substitute any values that could be used to identify the user with different values.
• Suppression—Omit (all or in part) any values that could be used to identify the user.
• Generalization—Substitute specific values that could be used to identify the user with something less specific. For example, generalize the date of birth to the year or decade in which the user was born.
• Perturbation—Make random changes to the data to corrupt values that could be used to identify the user.
• Anonymize non-sensitive data as well, if it could be used for the reverse anonymization of sensitive data.
• Make sure that the masking process is not reversible.
• Make sure that the same masking process will produce the same results each time.
• Make sure that data types remain compliant with the schema.
• Preserve the meaning of the data.

Delete Private or Sensitive Data That is No Longer Needed:
Defects make the software fail to delete private or sensitive data that is no longer needed, putting privacy at risk. To avoid this defect:
• Design the software to minimize data that is stored in the first place.
• Promptly delete data that is no longer needed.
• Properly delete data when a user issues a rightful request.
• Securely lock the data from any access until deletion is possible, if prompt deletion is not possible due to technical restrictions.
• Ensure prompt deletion of data in backups, copies, cloud storage, or data shared with third-party sources.
• Clearly inform users when backups must be kept, as required by law.
• Provide evidence (such as logging and messaging to the user) to verify deletion according to policy.
• Identify deletion policies (circumstances under which data must be deleted, and the timeframe for deletion), and implement automation and/or manual procedures to ensure that happens.

Make Sure Privacy Policies, Terms and Conditions are Clear:
The software may not make it clear to users what it will do with their data so users can make good decisions about how to manage their data within the software. To avoid this defect:
• Provide release notes with software updates to clearly and simply explain how terms and conditions change over time.
• Track which users have consented to the terms and conditions, including the version if terms and conditions have changed over time.
• Implement a Do Not Track feature on the server side, so users can disable tracking, and provide an opt-out capability for users.
• Provide users with a list of all tracking mechanisms used in the software, explaining how and by whom the information is used.
• Inform users (through a clear and well-written terms and conditions page, for example) how data is processed, including collection, storage, processing, and deletion.

Do Not Collect Non-Essential Data:
When data that is not needed to meet requirements is collected, it needlessly puts privacy at risk. To avoid this defect:
• Do not collect descriptive, demographic, or any other user-related data that are not needed for the purposes of the system.
• Enable users to opt out of providing additional data to improve the service.

Do Not Share Data without Consent:
The software should not provide data to a third party without obtaining the user's consent. To avoid this defect:
• Acquire and document the user's consent for any data collected before the data is actually collected.
• Acquire and document the user's consent for any additional data that is collected later (due to software feature updates or new compliance requirements, for example).
• Mark web requests as Do Not Track, complying with the latest W3C standards.
• Anonymize personal data before sharing it with a third party.
• Do not share data inadvertently by embedding third party resources such as third party hosted JavaScript, JavaScript widgets, analytics components, advertisements, and so forth.