Automation and Orchestration

Objective 4.7: Explain the importance of automation and orchestration related to secure operations

- **Automation and Orchestration**
    - *Automation*
        - Execution of tasks without manual intervention
        - Purpose
            - Consistency, efficiency, reduction of human error
        - Example
            - Scripting repetitive tasks
    - *Orchestration*
        - Coordinated execution of multiple automated tasks for a specific outcome or workflow
        - Purpose
            - Ensures tasks work together harmoniously
        - Example
            - Sequencing tasks in incident response
    - *SOAR (Security Orchestration, Automation, and Response)*
        - Class of security tools for incident response, threat hunting, and security configurations
        - Purpose
            - Orchestrate and automate runbooks, deliver data enrichment
        - Example
            - Integrating SIEM and SOAR for advanced security capabilities

- *Playbook*
  - Checklist of actions for detecting and responding to a specific incident
  - Role
    - Guides incident response processes
  - Example
    - Steps for responding to a phishing campaign
- *Runbook*
  - Automated version of a playbook with defined interaction points for human analysis
  - Role
    - Executes automated tasks with human decision points
  - Example
    - Automated incident response with analyst decision points
- Benefits of Automation and Orchestration
  - Efficiency
    - Time-saving and consistent execution
  - Standardization
    - Enforces baselines and standardized configurations
  - Scalability
    - Scales securely and efficiently
  - Employee Retention
    - Reduces repetitive tasks
  - Reaction Time
    - Faster responses to incidents
  - Workforce Multiplier
    - Maximizes human resources

- **When to Automate and Orchestrate**
  - *Automation and Orchestration*
    - Effective automation and orchestration are for repeatable and stable tasks and workflows
    - Identify consistent processes in your organization for automation and orchestration
  - Decision factors for implementing automation and orchestration
    - *Complexity*
      - Automation and orchestration are suitable for complex, repetitive tasks
      - Determine process complexity to decide whether to automate or orchestrate
      - Routine backups are suitable for automation, while complex incident response requires orchestration
    - *Cost*
      - Initial investment is a key factor
      - Conduct a cost-benefit analysis considering development, implementation, and maintenance costs
      - Include hardware, software, personnel, and support costs in the analysis
      - Cost savings often outweigh the initial investment in the long run
    - *Single Points of Failure*
      - Implement backup systems or manual processes to mitigate single points of failure
      - Redundancy and failover mechanisms, both technical and manual, can ensure uninterrupted operations

        - *Technical Debt*
            - Technical debt is the cost and complexity of suboptimal software solutions
            - Regular reviews and updates are necessary to avoid technical debt
            - Technical debt can impede efficiency and security
        - *Ongoing supportability*
            - Automation and orchestration systems need ongoing maintenance and adaptation
            - Teams must possess the necessary skills to maintain and adapt these systems
            - Training and skill development are essential
            - Most automation depends on the connection of systems via APIs and webhooks

- **Benefits of Automation and Orchestration**
    - Increased Efficiency and Time Savings
        - Automation reduces manual tasks
        - Repetitive processes, like patching and backups, can run seamlessly without human intervention
        - Frees up human resources and reduces the risk of errors
        - Increases reliability and consistency in processes
    - Enforcement of Baselines
        - Consistently enforces security and compliance baselines
        - Defines standardized configurations and policies
        - Ensures systems align with industry best practices and regulatory requirements
        - Minimizes vulnerabilities and security breach risks

- Implementation of Standard Infrastructure Configurations
  - Facilitates the creation and enforcement of standard configurations
  - Ensures consistent setup of all systems
  - Detects deviations from established standards and triggers automated corrective action
- Secure Scaling
  - Enables secure scaling of IT infrastructure as organizations grow
  - Dynamically scales resources while adhering to security protocols
  - Provisioning virtual machines, adding network resources, and access control adjustments are done securely
- Increased Employee Retention
  - Empowers employees to focus on strategic and creative aspects of their roles
  - Reduces repetitive and mundane tasks
  - Increases job fulfillment and engagement
  - Reduces the risk of burnout, contributing to higher retention rates
- Faster Reaction Times
  - Facilitates rapid response to security incidents and threats
  - Automation and orchestration systems are always available
  - Automates intrusion detection, threat analysis, and incident response
  - Real-time alerts and predefined response actions enhance security
- Workforce Multiplier
  - Augments existing staff's capabilities
  - Smaller teams can manage larger, more complex infrastructures
  - Reduces staffing needs and optimizes resource allocation for cost savings

- **Automating Support Tickets**
  - Automating Support Ticket Management
    - Enhances IT and customer support team efficiency
    - Streamlines issue resolution and improves customer satisfaction
    - Support ticket management is critical for addressing issues, incidents, and service requests
    - High ticket volume can lead to delays, increased workloads, and decreased customer satisfaction
  - Automating Support Ticket Creation
    - Six steps in the ticket creation process
      - Users submit requests through channels (e.g., email, web form, support portal)
      - Automation tool generates tickets based on predefined criteria
      - Capture essential information from user submissions
      - Categorize tickets based on content or source
      - Assign priority based on predefined rules and criteria
      - Automated notification sent to relevant support team or technician
    - Benefits of Automating Ticket Creation
      - Ensures efficient capture, categorization, and assignment of support requests
      - Reduces the risk of lost or overlooked tickets
      - Accelerates response time to user needs
  - Ticket Escalation Automation
    - Ticket escalation addresses complex or high-priority issues
    - Follows a five-step process
      - Define escalation criteria based on issue nature, urgency, and service

            level agreements
            - Create automation rules to monitor ticket attributes and trigger escalation
            - Perform predefined escalation actions (e.g., notification, reassignment, change in priority)
            - Monitor and track the escalated ticket's progress
            - Resolve and close the ticket, triggering notification to the user
        - Benefits of Automating Ticket Escalation
            - Ensures prompt handling of critical issues
            - Maintains transparency and accountability in the support process
            - Helps meet service level agreements and minimize delays in addressing urgent matters

- **Automating Onboarding**
    - *Automation*
        - Involves using technology to execute repetitive tasks without continuous human intervention
    - Automating the onboarding process impacts organizational productivity, employee satisfaction, and retention rates
        - Streamlining onboarding ensures new hires are integrated quickly and efficiently into their roles and the organization's culture
        - Benefits
            - Eliminates manual tasks, reduces errors, and provides structured, consistent onboarding
            - Reduces administrative burden on HR and IT departments
            - Enhances support ticket management processes

- Areas to Automate in Onboarding
  - Creation of documentation records
  - Scheduling training
  - Provisioning equipment
  - Managing access rights
  - Distributing checklists
  - Collecting feedback
- *User Provisioning*
  - Involves creating and managing user accounts and access rights
  - Ensures new employees have necessary access to systems, applications, and resources
  - Process includes the following
    - Collecting information
    - Creating accounts
    - Assigning roles and access
    - Sending notifications
    - Conducting synchronization and updates
  - Steps in User Provisioning
    - Employee provides personal details, role, and department information
    - Automation creates user accounts in various systems
    - Automation assigns roles and access levels based on department and position
    - Automated notifications sent to the employee, manager, or IT department
    - Automation keeps user information synchronized across platforms

- *Resource Provisioning*
  - Ensures timely allocation of physical and digital resources needed by new employees
  - Resources include
    - Workstations
    - Software licenses
    - Communication tools
  - Process involves
    - Requirements analysis
    - Resource allocation
    - Configuration
    - Verification and auditing
    - Gathering feedback
  - Steps in Resource Provisioning
    - Analyze role and department information to determine specific resources
    - Initiate procurement workflows or allocate available resources based on rules
    - Configure resources to meet the employee's role
    - Verification process to ensure successful allocation
    - Auditing to track allocated resources for inventory management and compliance
    - Employee and manager feedback on resource suitability and additional requirements

- **Automating Security**
  - Automating Security
    - Helps prevent security vulnerabilities, respond to threats swiftly, and maintain consistent security policies
    - It involves using technology to perform crucial but repetitive security tasks to maintain updated defenses and swift response to security threats
    - Automation includes the use and configuration of guardrails, security groups, service access management, and permissions
  - Ways to Automate Security
    - Implementing Guardrails
      - Guardrails are automated safety controls to protect against insecure infrastructure configurations
      - Configured according to security standards and enforce security policies automatically
      - Continuously monitor infrastructure, detect security violations, and take predefined corrective actions
    - Managing Security Groups
      - Security groups act as virtual firewalls for cloud-based server instances
      - Specify allowed incoming and outgoing network traffic using predefined rules
      - Automate assignment of instances to appropriate security groups
      - Dynamically adjust security group configurations to respond to evolving threats
      - Analyze traffic for unauthorized access attempts
    - Enabling and Disabling Services and Access
      - Automate service access management to prevent unnecessary risks and

maintain operational efficiency

- Regularly review and manage access to services
- Monitor for unusual activity and automatically restrict or disable access if suspicious
- Enable or disable services based on a predefined schedule when not continuously needed

- Automating Permissions Management
    - Manage permissions using Role-based Access Controls (RBAC)
    - Automate provisioning and de-provisioning of access rights based on assigned roles
    - Ensure no unauthorized access to sensitive information
    - Perform regular checks on permissions settings to verify compliance with policies and regulations
    - Make necessary adjustments over time to maintain security

- **Automating Application Development**
    - Automating Application Development
        - Enhances efficiency, consistency, and the quality of software products
        - *Automation*
            - In application development, it involves using technology to manage, test, and deploy applications with minimal human intervention
    - Continuous Integration and Continuous Deployment (CI/CD) significantly improve software efficiency, consistency, and quality
        - *Continuous Integration (CI)*
            - Developers merge code changes frequently in a central repository
            - Automated build process verifies each check-in and detects problems

        during integration
    - Automation tools manage code integration, provide notifications for conflicts or errors
    - Automated tests ensure software quality after integration
    - Developers receive feedback on detected issues to make necessary corrections
    - *Release*
        - Process of finalizing and preparing new software or updates
        - Enabling software installation and usage
    - *Deployment*
        - Involves automated process of software releases to users
        - Actual installation of software in a new environment
- *Continuous Integration and Continuous Delivery (CI/CD)*
    - CI/CD includes continuous integration
    - Continuous Delivery (CD) ensures code is always deployable after every change
        - Automated testing and build processes
        - CD stops short of automatic production deployment
        - CD is part of the release process
        - Full deployment process is automated only to a certain stage
            - Doesn’t deploy into the production environment automatically
        - Deployment to production environment is a manual business decision
        - Allows flexibility in timing, market conditions, and stakeholder readiness

        - *Continuous Deployment*
            - Takes CI/CD further by automatically deploying code changes to testing and production environments
            - All changes passing through the production pipeline are fully released with no human intervention
            - Automation ensures consistent deployments, faster releases, and offers rollback capabilities
            - Requires a paradigm shift, more developer involvement in the deployment process
            - Promotes increased communication and collaboration within teams for collective responsibility
        - Benefits of CI/CD
            - Adapting to changing market demands more quickly
            - Efficient workflow from development to deployment
            - Improves code quality, streamlines deployment processes, and allows flexible production release
            - Reduces deployment risks and enhances software reliability

- **Integrations and APIs**
    - *Integration*
        - Combining subsystems or components into a single, functioning system
    - *API (Application Programming Interface)*
        - Set of rules and protocols used for building and integrating application software
        - Enable software developers to access functions or features of another application programmatically

- API Communication
  - APIs facilitate communication between different parts of a microservice or service-oriented architecture
  - Allows automation of administration, management, and monitoring of services and cloud-based infrastructures
  - Common communication methods used by APIs
    - *REST (Representational State Transfer)*
      - REST uses standard HTTP methods, status codes, URIs, and MIME types for interactions
      - Primarily uses JSON for data transfer
      - Lightweight protocol suitable for integrating with existing websites
    - *SOAP (Simple Object Access Protocol)*
      - SOAP has a structured message format in XML
      - Known for robustness, additional security features, and transaction compliance
      - Suitable for enterprise-level web services with complex transactions and regulatory compliance requirements
- Benefits of API Integrations
  - Improved efficiency and consistency
  - Allows direct integration of third-party applications into web applications
  - Reduces the need to build entire services from scratch
- API Testing with CURL
  - *CURL*
    - A tool for transferring data to or from a server using various supported protocols
  - Commonly used protocols for API testing are HTTP and HTTPS

- Use CURL to send data to an API and receive a response for testing
- CURL allows sending data to an API and receiving a JSON response
- Helpful for software developers and cybersecurity professionals, especially in penetration testing scenarios