**Group Project Assignment: Designing an Organizational Cybersecurity System**

**Project Overview:**

In this project, you will work in groups to design and develop a comprehensive cybersecurity system for a hypothetical organization. The goal is to apply the concepts and techniques learned throughout the course, including systems engineering principles, UML modeling, Python automation, data analytics, and risk management. The project will challenge you to consider the various aspects of cybersecurity, from system architecture and implementation to monitoring and incident response.

**Objectives:**

* Apply systems engineering concepts to design a robust cybersecurity system.
* Utilize UML to model the system architecture and processes.
* Implement Python automation scripts for various cybersecurity tasks.
* Integrate data analytics and machine learning for threat detection and response.
* Develop risk management and disaster recovery plans.

**Project Timeline:**

* **Week 5:** Introduction to the project, group formation, and initial planning.
* **Weeks 6-8:** System design and modeling using UML/SysML.
* **Weeks 9-10:** Python scripting for automation and basic data analytics.
* **Weeks 11-12:** Advanced data analytics and machine learning implementation.
* **Week 13:** Risk management and disaster recovery planning.
* **Week 14:** Final integration and testing.
* **Week 15:** Project presentation and submission.

**Deliverables:**

1. **System Design Document (Due Week 8):** A detailed report including UML/SysML diagrams that illustrate the system architecture, components, data flow, and security processes.
2. **Python Scripts and Automation (Due Week 10):** Scripts for automating tasks such as log analysis, system monitoring, and basic threat detection.
3. **Data Analytics Report (Due Week 12):** Analysis using real or simulated data to demonstrate how the system can identify and respond to security threats.
4. **Risk Management Plan (Due Week 13):** A document outlining the system's risk management strategy, including incident response and disaster recovery plans.
5. **Final Project Presentation and Submission (Due Week 15):** A presentation summarizing the system design, implementation, and findings. Submit all documentation, code, and presentation slides.

**Detailed Breakdown and Requirements:**

**Week 5: Project Kickoff and Group Formation**

* **Activity:** Introduction to the project, group formation, and topic discussion.
* **Output:** Each group will select an organization type (e.g., financial, healthcare, government) and outline the scope of their cybersecurity system.
* **Tasks:**
  + Form groups of 4-5 students.
  + Define the organization’s size, industry, and typical security challenges.
  + Create a project plan with milestones and assign roles within the group.

**Weeks 6-8: System Design and Modeling**

* **Activity:** Design the cybersecurity system using UML/SysML diagrams.
* **Output:** System Design Document.
* **Tasks:**
  + Use UML to create use case diagrams, activity diagrams, and class diagrams to model system requirements and user interactions.
  + Use SysML to develop block definition diagrams (BDD) and internal block diagrams (IBD) to detail system components and their interactions.
  + Develop sequence diagrams to show the flow of data and control within the system.

**Weeks 9-10: Python Scripting and Automation**

* **Activity:** Implement Python scripts to automate cybersecurity tasks.
* **Output:** A set of Python scripts and a brief report explaining their functionality.
* **Tasks:**
  + Create scripts for log file analysis, system performance monitoring, and alert generation.
  + Develop automation for routine security checks, such as scanning for vulnerabilities and monitoring network traffic.
  + Include error handling and logging to ensure reliability.

**Weeks 11-12: Data Analytics and Machine Learning**

* **Activity:** Integrate data analytics and machine learning into the system.
* **Output:** Data Analytics Report and Python code.
* **Tasks:**
  + Collect and preprocess data (real or simulated) relevant to the organization's security posture.
  + Apply data analytics techniques to identify patterns and anomalies in the data.
  + Implement basic machine learning models (e.g., anomaly detection) to enhance threat detection capabilities.
  + Visualize findings using plots and dashboards.

**Week 13: Risk Management and Disaster Recovery**

* **Activity:** Develop a risk management plan.
* **Output:** Risk Management Plan document.
* **Tasks:**
  + Identify potential risks and vulnerabilities within the system.
  + Develop a risk assessment matrix to prioritize threats.
  + Outline incident response procedures, including detection, response, and recovery.
  + Create a disaster recovery plan addressing data backup, system restoration, and continuity of operations.

**Week 14: Integration and Testing**

* **Activity:** Integrate system components and conduct testing.
* **Output:** Integrated system and test report.
* **Tasks:**
  + Integrate all components of the system, including scripts and analytics tools.
  + Perform testing to ensure system functionality, including unit testing, integration testing, and security testing.
  + Document any issues encountered and how they were resolved.

**Week 15: Project Presentation and Submission**

* **Activity:** Present the final project to the class.
* **Output:** Presentation and final submission.
* **Tasks:**
  + Prepare a presentation that includes an overview of the system design, implementation, and key findings.
  + Demonstrate the functionality of the system with a live or recorded demo.
  + Submit all project documentation, code, and presentation slides.

**Grading Criteria:**

* **System Design and Modeling (25%):** Clarity, completeness, and correctness of UML diagrams.
* **Python Scripting and Automation (20%):** Functionality, reliability, and documentation of scripts.
* **Data Analytics and Machine Learning (20%):** Depth of analysis, accuracy of models, and insights gained.
* **Peer Evaluation (15%):** Feedback on participation and contribution from Team Members.
* **Presentation and Integration (10%):** Quality of the final presentation, system integration, and testing.
* **Collaboration and Effort (10%):** Teamwork, meeting milestones, and overall effort.

**Additional Notes:**

* Each group should have a designated project manager to coordinate tasks and ensure deadlines are met.
* Regular check-ins with the instructor will be scheduled to monitor progress and provide guidance.
* Consider using version control (e.g., Git) to manage code and document changes.