Part 1: Computer and Software Fundamentals

1. **Basic Computer Architecture**
    a. What is a CPU and how it processes instructions
    b. Role of memory (RAM), registers, stack, and heap
    c. Concept of the call stack and function calls
2. **Operating System Basics**
    a. What are processes and threads
    b. How memory is managed in programs
    c. Introduction to file systems and I/O operations
    d. Understanding synchronization and race conditions at a high level
3. **Software Compilation and Execution**
    a. How source code becomes a program (compile → link → run)
    b. Static vs dynamic linking (conceptual)
    c. Basics of executable files (just high-level formats like ELF, PE)
4. **Programming Language Concepts (C/C++)**
    a. How memory is allocated (stack vs heap)
    b. Common mistakes in C/C++ (e.g., uninitialized memory, pointer errors)
    c. Concept of undefined behavior

Part 2: Common Software Vulnerabilities

5. **Memory-Based Vulnerabilities**
    a. What is a buffer overflow?
    b. Understanding format string issues
    c. Concept of use-after-free and double free
    d. Integer overflows and off-by-one errors
6. **Race Conditions**
    a. Simple explanation of concurrency problems
    b. What is a time-of-check vs time-of-use (TOCTOU) issue?

Part 3: Understanding Vulnerability Discovery Techniques

7. **Concept of Static and Dynamic Analysis**
    a. What is static analysis?
    b. What is dynamic analysis?

c. High-level benefits and limitations of each

8. **Overview of Analysis Techniques**
    a. What is data flow and control flow?
    b. What does "symbolic execution" mean ?
    c. What is taint analysis?

Part 4: Fuzzing – Conceptual Overview

9. **Introduction to Fuzzing**
    a. What is fuzzing and why it's used
    b. Difference between dumb and smart fuzzing
    c. How input mutation and generation work
10. **Grammar-Based Fuzzing**

- What is a grammar?
- Why structured inputs are important

11. **Theory Behind Fuzzing Strategies**

- What is coverage-guided fuzzing?
- Importance of code coverage in fuzzing

Part 5: Understanding Exploitability and Mitigations

12. **Crash Concepts**

- What is a crash and why it matters
- Simple view of how crashes help discover bugs

13. **Common Security Defenses**

- What is DEP (Data Execution Prevention)?
- What is ASLR (Address Space Layout Randomization)?
- What is a Stack Canary?
- What is SEH (Structured Exception Handling)?

14. **Vulnerability Scoring Basics**

- What is CVSS and why it's important
- Basic explanation of impact, complexity, and vector

Part 6: Ethical Aspects of Vulnerability Research

15. Responsible Disclosure

- What is ethical hacking?
- Coordinated vulnerability disclosure
- Importance of ethics in cybersecurity research