Exercises 14.1, 14.5, 14.7, 14.8

14.1. Explain how the complementary strategies of resistance, recognition, recovery, and reinstatement may be used to provide system resilience.

System resilience means keeping services running even when stuff breaks or gets attacked.

Recognition: Spot early signs of failure or attack fast.

Resistance: Stop or limit damage using defenses like firewalls or isolation

Recovery: Get key services back quick using backups or spare systems.

Reinstatement: Fully restore everything so normal ops continue.

All four work together to keep the system tough and steady.

14.5. What is survivable systems analysis and what are the key activities in each of the four stages involved in it as shown in Figure 14.8?

Survivable systems analysis finds vital system parts that could be hit by attacks and builds ways to keep them running.

1. System understanding: Know what the system does and how it's built.
2. Critical service identification: Find must-have services and components.
3. Attack simulation: Imagine attack cases and what parts get hit.
4. Survivability analysis: Pick weak but vital parts and plan defense using resistance, recognition, and recovery.

14.7. Suggest how the approach to resilience engineering that I proposed in Figure 14.9 could be used in conjunction with an agile development process for the software in the system. What problems might arise in using agile development for systems where resilience is important?

Resilience ideas like planning for the four R's can fit into agile by adding security and recovery work into each sprint. But agile's fast pace and weak docs can make it hard to plan deep resilience stuff that needs long-term design and solid requirements.

14.8. In Section 13.4.2, (1) an unauthorized user places malicious orders to move prices and (2) an intrusion corrupts the database of transactions that have taken place. For each of these cyberattacks, identify resistance, recognition, and recovery strategies that might be used.

1) Malicious orders attack

Recognition: Watch logs for weird or high-volume order activity.

Resistance: Use strong user checks and isolate price systems.

Recovery: Restore correct prices using clean data backups fast.

2) Database corruption attack

Recognition: Use auto integrity checks and alerts for data changes.

Resistance: Protect database with firewalls, encryption, and isolation.

Recovery: Reload from backup and replay safe transactions to fix it.