Written questions:

1b. Compromised CIA properties: Availability. The freezing of Jane's computer limits the availability of her services and data. This compromises availability.

Possible attack method: Ransomware attack. Jane could have visited a malicious website that contained ransomware. The attacker is likely to ask for payment to decrypt or unfreeze her data.

1c. Compromised CIA properties: Confidentiality. If Jane is correct in that there is a location tracker on her phone, her private data such as location and movements are being monitored and tracked by an external party that isn't authorized. This compromises Jane's confidentiality.

Possible attack method: Jane could have potentially connected to an unsecured or compromised public wifi network. Attackers might have intercepted her traffic and used it to deliver spyware to her device.

2a. Both.
Security: Attackers might make use of the vulnerability to alter or gain unauthorized access to her smartphone data through this car sharing feature. If done successfully, both confidentiality and integrity would be compromised as Jane's data could be altered without her knowing.
Privacy: Attackers or other users now potentially have access to Jane's sensitive personal data. They may access private emails, messages, or locations. This violates Jane's privacy.

2b. 1.Check allowed permissions for multi user features. If Jane believes that the isolation is not properly done, she could disable some of the authorisation for more private data. For example, she could disallow the car from accessing and connecting to her messages. While this would reduce the utility of the car, it is still a step worth taking if she believes there are vulnerabilities.

2. Only share the car with trusted individuals, and do not 'cross contaminate'. Jane currently shares the car with both family members and colleagues, and connects her smartphone which contains both personal and professional data. If possible, she should limit who shares the car with her, and try to only allow either private users, or professional users.

3a. Place limits to filter incoming traffic and block suspicious or excessive requests.

3b. Security policies should be clearly communicated and well defined. This should be made publicly known so that attacks are aware of how robust the system is, deterring them from trying to attack.

3c. Use a system to direct traffic to the nearest available data center. If one server is under attack, traffic can be redirected to other unaffected servers. Attackers that are made aware of this fact are more likely to go after servers that do not make use of this system.

3d. Monitor network traffic and detect any unusual or sudden increase of traffic. If it occurs, raise an alert and immediate action can be taken.

3e. Incident response plan. Develop a detailed plan that has what steps to take in the event of a DoS attack. This includes protocols and strategies to reduce downtime such as rerouting traffic.

4b. Ransomware. Mainly spread through phishing emails. Once a system is infected, Conti will encrypt the user's data and information. They require a ransom in order to decrypt the information. If no payment is made, the user's sensitive data is also threatened to be released.

4c. Spyware. Enters the machine by being secretly and remotely installed on mobile devices. For iOS devices, it was installed through 16.0.3 using the zero clock exploit. Once on the machine, it can snoop through personal information like messages, passwords, phone calls and emails.

4d. Worm, Trojan. Used to obtain passwords and banking related data. Enters machines by self propagating over to different systems. It is used to steal private information like passwords and banking information. Attackers can also have follow up actions that can be executed remotely.

4e. Logic Bomb. Designed to attack Iran's centrifuge system, stuxnet entered the machine through external removable drives. It exploits zero day vulnerabilities in Microsoft Windows, causing the centrifuges to spin out of sync and ruining the production.

Programming question exploit descriptions:

Buffer overflow vulnerability
In pwgen.c we observe the vulnerability in print_usage function. The vulnerability occurs in the strncat(). The linux IA32 machine, it has a stack with argv variables, environ variables followed by the stack. The code has a buffer with 1024 bytes. We set the argv[] and environ[] in the stack to null values, followed by setting the buffer to fill up with 1024 bytes worth of characters A. memcpy the shellcode 45 times. We then memcpy to the address which we calculate by using gdb. We calculate that the memory address is 1024 + 45 ("At least one option …. ") bytes away. By calculating the stack and argv variables we deduce the memory address, we store the exploit_addr inside the argv[0] variable and we use the return address of print_usage to call the commands that executes the new shellcode.

Environment variables vulnerability
The exploit code attempts to change the HOME environment variable and escalate privileges to the root user. We use setenv to set the home environment to root user directory and using pwgen system to generate a password. We store the generated password and switch to root user using su command and provide the password.

Permissions vulnerability

Inside the update_spent function, there is a possible permissions vulnerability. We observe this by looking at the check_perms() function and realise it can write to the file as root but does not check if current user writing is the owner of the file. Fill_entropy() function also does not do any permission checking. Our exploit code exploits this vulnerability by creating a symbolic link. Our exploit file removes the original file created by pwgen program and we create a new symbolic link and we modify that new file with our exploit string which are commands for the root user to create a new ssh and define another new user. This will cause the shellcode to be executed and create a new shell for us thus making the exploit successful.