Cordelia Tan
C62tan@uwaterloo.ca
21112180
CS458 System Security Assignment 1

1b. Compromised CIA properties: Availability. By preventing Jane from accessing her computer, they are compromising the availability of her data.
Possible attack method: Display of fake error messages. Attacks might have created fake pop ups to mimic error messages and instruct Jane to call their number

c. Confidentiality. If a location tracker is installed on her phone, this compromises confidentiality as it reveals sensitive information about her whereabouts.
Possible attack method: Spyware apps. Jane could have installed a malicious app and granted it location access on her phone. This could have been made use by the attacker.

2a Both. Privacy - Her personal data may be leaked including her addresses, location, etc. This is a privacy issue as personal information will be made known to others other than Jane. It is also a security concern as it compromises confidentiality, attackers could also potentially make use of this vulnerability to exploit the car system and thus Jane's system.

b. Regularly update the car system and her smartphone. They may be potential threats or vulnerabilities related to this feature on the car system/ smartphone that may be discovered. As such, patches may be released and Jane should immediately update this once it's released to ensure that attackers do not use that vulnerability to exploit her system

Remove permissions from the car system. Jane could also remove permissions regarding certain sensitive information to prevent the feature from accessing it. This would ensure that in the event of a leak, those sensitive data would not be released.

3a. Limit the number of requests from a single source.

b. The company could enforce strict legal consequences that warns potential attackers of legal repercussions.

c. Utilize Content delivery networks. Make use of these networks to cache, store and serve static content for the website instead. It can also absorb a lot of the network load for the origin server. This would deter attackers as they know that a DoS attack would not be effective.

d. Network monitoring. Track the number of requests coming from sources, raise an alert if they detected anything suspicious

e. Have redundancy infrastructure that allows the system time to recover in the event of an attack. While one portion of the system is down, the redundancy part of the system can handle the load.

4b. Ransomware. Conti is spread through phishing emails or compromised websites. Once infected, it will start to encrypt the users data and demand a ransom in order for the user to get their data back. There’s also a double extortion technique where they threaten to release sensitive data is payment is not made.

c. Spyware. Designed by Israeli cyber arms company that is meant to be covertly and remotely installed on mobile phones running iOS and Android. They are able to install Pegasus on iOS devices through 16.0.3 via the zero clock exploit. Once installed, it is able to read text messages, collect passwords, snoop on phone calls and generally collect sensitive and private data from the phone.

d. Trojan, worm. Mainly a banking Trojan and password stealer. Spreads by self propagating to other systems. It is able to steal sensitive data such as passwords, financial data, stored emails, etc. It also provides remote code execution to allow attacks to perform secondary actions.

e. Logic bomb, worm. Stuxnet was developed to target Iran's nuclear program. Propagated using removable drives, it exploits zero day vulnerabilities in Microsoft Windows. It can manipulate the centrifuges systems, causing them to spin erratically.

PART B write up

Programming questions

Sploit2.c
Environment variable vulnerability
The exploit code aims to modify the HOME environment variable to gain root-level access. It accomplishes this by utilizing the setenv function to change the home environment to the root user's directory. Subsequently,it calls the pwgen system call to generate a password and save it to a file called p.txt. This generated password is saved, allowing the code to transition to the root user's privileges using the su command while providing the newly generated password for authentication.

Sploit3.c
Buffer overflow vulnerability

In the pwgen.c code, a vulnerability can be identified within the print_usage function. This vulnerability revolves around the use of strncat function when the argument count is less than 2. Although there is a check for the buffer, however we are still able to overflow the buffer as the check allows us to overflow 1024 + 45 bytes. On a Linux IA32 machine, the stack typically contains argv and environ variables, and then additional stack data. In the code, there's a buffer allocated with a size of 1024 bytes.

We manipulate the argv[] and environ[] variables on the stack, setting them to null values. Next, we fill the buffer with 1024 bytes of rubbish characters. We replicate the shellcode 45 times using the memcpy function. The critical step is determining the memory address to which we want to copy our shellcode. This address can be calculated using debugging tools like GDB. Specifically, we calculate that this memory address is 1024 + 45 bytes away from the start. We store this calculated exploit address inside the argv variable which is at the top of the linux IA 32 machine. We leverage the return address of the print_usage function to execute commands that trigger the new shellcode.

Sploit4.c
Permissions vulnerability
The update_spent function exhibits a potential permission vulnerability. This vulnerability becomes apparent when examining the check_perms function, which is capable of writing to files as the root user without verifying whether the current user possesses ownership of the file. Furthermore, the fill_entropy() function lacks any permission checks.

Our exploit code capitalizes on this vulnerability by generating a symbolic link. In this exploit, we remove the /tmp/random_pwgen file generated by the pwgen program and replace it with a new symbolic link. Subsequently, we modify this newly linked file with our exploit string, which contains commands intended for the root user. These commands are designed to establish a new SSH connection and define an additional user account. As a result, the shellcode embedded within our exploit is executed, leading to the successful exploitation of the system. This enables us to create a new shell, providing us with unauthorized access.