cs458 A1

j585zhan

October 2018

1 Writing Part

Q1
a. This scenario is a compromise of the privacy. Privacy means the data owner has the control to who can see the data and who the data can been given to. However in this case, data owners are obviously not permit Google to provide their personal information.
b. In this scenario, it is a compromise of the availability. On the FBI's point of view, the system/data is not there whenever they want it after the shutting down. This opposite to the availability in CIA/Privacy.
c. This scenario is also a compromise of the privacy. In this case, users of the phone companies lost the control of 'who get to see it'. There should be only the caller and phone call receiver get to access the phone call content, however FBI also get to access it, and this action is not permitted by users.
d. This is a scenario that compromise of the data integrity. Data integrity promises the accessed data is always correct. However when the hacker removes her team's data from the database, the database's information is not correct anymore.

Q2
a. This is an interruption. The blacklisted IP address cannot access Google's data anymore due to their access is interrupted.
b. This is an interception. The phone call should go from the caller to the phone call receiver. However, in the middle, FBI intercept the call, and makes recorded of the users activity.
c. This is an example of fabrication. The profile that FBI shows to thieves is fictional. They make it up and try to use it to convince the thieve.
d. This is a modification example. The reason of that is because FBI changes the original detail of the communication to something else.

Q3
a. Deflecting: Leave the electronic footprint/signature(name or icon) from another hack group. FBI will think it is done by another group and investigate on them.
b. Detecting: Install anti-monitoring software to the cellphone. If I found my

phone being monitored by stranger, I would get alarmed.
c. Recovering: I can laundering the stolen money to other country(backup). I can still access the money after I get out from jail one day (if possible).
Q4
a. Kovter is a Trojan, ransomware and logic bomb. First stage it is a ransomware. It wait in the system until user does specific action. The trigger is when user download the illegal files from internet, it will ask for "fine". Later, it infects computers by spam emails. When user click on the attachment of the spam email, it will trigger the Macro functions defined in the Microsoft file. The code will spawn a power shell, and download other part of the malware. The malicious code will steal personal information, and download more malicious code to the computer.
$https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/kovter-an-evolving-malware-gone-fileless$
b. ILOVEYOU is a worm. It infects computer by sending love letter in the email attachment, with the subject line "ILOVEYOU". It is a visual basic script. Once user open that, the malware will searching for the username/password stores on the machine and send to the malware developer. At the same time, the malware will be distributed by the outlook contact. It is also a trojan, because users made their own decision to open the malware. The subject line attract users to open it.
$http://malware.wikia.com/wiki/ILoveYou$
c. Mirai is also a worm. It infects other computers by scanning the network service like telnet. Once there are enough infected computers, it will use DDoS attack to the targeted server.
$https://en.wikipedia.org/wiki/Mirai_(malware)$
d. NotPetya is a ransomware and worm. It uses similar strategy as EternalBlue, scanning network and try to sploit any computer that is vulnerabile to the Server Message Block, and infects them by encrypting all of the files, unless send 300 dollar to the malware developer.
$https://www.csoonline.com/article/3233210/ransomware/petya-ransomware-and-notpetya-malware-what-you-need-to-know-now.html$

2 Programming Part

Sploit3
Vulnerability: This is an incomplete mediation attack.
Exploit: The program uses execvp find when doing 'backup ls'. However the execvp and using the relative path to invoke command path, makes the execution depends on the PATH environment variable. And if PATH is not in the environment variable, it will include current directory. By writing a script called 'find' which simply spawn a shell, and give it execution privilege to all of the users(chmod 777), execvp will actually execute this user script find instead of the find under /usr/bin directory. When backup doing execvp, it have root

previllege, so the spawned shell also has the root previllege.
Fix: user absolute path to invoke the command path, instead of dependes on the PATH variable.

Sploit4
Vulnerability: This is a TOCTTOU attack.
Exploit: between the program check file's permission and do the execution of restore command, there is no lock to guarantee the file is remaining unchanged in between. So if there are different threads running, some of them are creating the symlink to /etc/passwd to the file under current directory, and others just do backup. In some case, the file status will change during permission checking stage and restoring execution stage. And as a result, we might restore to the /etc/passwd due to the symlink.
Fix: add atomic lock to guarantee the file immutable between checking permission and do the operation. Or use the file handler.

Sploit5(Backdoor)
Vulnerability: This is a Backdoor attack. The vulnerability is off-by-one.
Exploit: By decompiling the backupV2, we found that around line 273, the strncpy changed to strncat. According to the manual for strncpy and strncat, strncat will write max in total n+1 bytes to the destinition however the strncpy will only write max n in total. As a result, we can overflow one byte of zero(string terminate char) to the stack. By checking the stack position of the ErrorMsg, I noticed that the variable permissionsError is next to the ErrorMsg. Which means, by overflow a single zero, we can change the permissionsError from 1(set permission error) to 0(no permission error). By doing that, we can bypass the permission check. To overflow the ErrorMsg, we can pass a super long path to backupV2. To use this vulnerability to get root privilege, we can first backup passwd under /etc directory, and put ./././ ..... ././passwd as the path, where the total length is greater than 152(minus the strlen of error msg). In this case we can restore a passwd written by user to /etc/passwd. If we put a root privilege account into our own passwd, and after restore to /etc/passwd, su to that account, we become the root.
Fix: if we use the strncat, we need to minus an extra 1 of the copying buffer size(for null character).