Swasti Mishra
Stella Sun
COSC 366
13 September 2022

COSC 366 Written Assignment 1

Question 1
Consider the following three scenarios:
A website that allows consumers to order products with their credit/debit card (e.g. Amazon).

(a) Who might want to attack the system?
Hackers may want to gain unlawful access to this site because a successful attack may yield credit card numbers, passwords that users probably use on multiple sites, customers' mailing addresses, email addresses, and a whole host of other information. This kind of information is valuable to all sorts of unsavory types- run-of-the-mill spammers would want it, more advanced criminals who may try to open accounts, order things, or otherwise funnel money out of accounts would want it, and even governments may want it.

(b) What types of harm might they want to cause?
See the above answer. Further, hackers may want to gain control of accounts to run things like botnets or mine crypto.

(c) What kinds of vulnerabilities might they exploit to cause harm?
Hackers could probably target whatever the login system is, but if the website is of any repute, they probably use a Single Sign On (SSO) service that is very difficult to crack. This means that hackers are far more likely to target people's behavior. For example, they may send people emails from accounts pretending to be said website saying, "Urgent Account Information Required," and then ask users to login on a realistic but false platform. When users fall for this scam, they hand bad actors their login information and potentially their credit card numbers as well.

A social media website that allows anyone to sign-up and post content.

(a) Who might want to attack the system?
Along with all the hackers for reasons described above, (namely email addresses and reused passwords), other individuals may want to attack the system as well. Governments may attack social media platforms to sway public opinion in their favor or subdue political protest on the site.

(b) What types of harm might they want to cause?
Besides the ones described above, other hackers may target specific popular accounts to post defamatory material or promote their personal interests.

(c) What kinds of vulnerabilities might they exploit to cause harm?
Besides the fake login page vulnerability, they may send users links saying things like "You have won a free iPhone!" that when clicked on, scrapes their information. Another vulnerability I've seen, particularly on platforms like Facebook Marketplace, is convincing users that they must send some money in order to get more money. A friend of mine nearly fell for a scam when she went to sell her Doc Martins, and an "interested customer" pretended that they were unable to send money because her account was configured in the wrong way. They then sent a fake email from PayPal that would "reconfigure her account"

for a small sum. They may also ask users to send their phone numbers and confirm a code, to confirm that a user is real, when in actuality, they are opening accounts in a different name. Most of these vulnerabilities seem to have less to do with actually breaking software, and more to do with tricking people into giving up their information.

An internet-connected thermostat that allows electric utility operators to adjust temperatures
to regulate power supply based on demand (e.g. when demand is high, operators turn off heating units 2).

(a) Who might want to attack the system?

The answers to this are far more diverse than the other two! Again, I think hostile governments may want to attack this system, especially if it is a Texas-like situation where people may potentially freeze to death in their homes because they don't have access to heating. Disgruntled homeowners may even try to attack the system if they aren't satisfied over their ability to control their own heating and cooling. It is also possible that larger corporations may attack the system to reroute more resources towards their ventures (similar to how golf courses are allowed to skirt water regulations in drought-stricken California).

(b) What types of harm might they want to cause?

See above.

(c) What kinds of vulnerabilities might they exploit to cause harm?

In this situation, the likelihood that the hackers try to use social engineering is pretty low, but not nonexistent. It's more likely that they target the customers' home networks or plant a bug in the electric utility's network that allows them to siphon off employee information.

Question 2

Suppose the course instructor has created a database of all information for this course: homework, exams, handouts, and grades. Create a detailed threat model for this database: what should the security goals be? What are reasonable attacks, and who are the potential attackers? What threats should we explicitly exclude from consideration?

Having a security goal requires identifying who the attackers are, and in this example, the hackers are very likely only students who are interested in gaining undue access to course material or manipulating their grades. The security goals should be to counteract these students' advances- essentially, protect the database from students who want to change their grade or access course material early. Reasonable attacks include accessing or unlocking the database, stealing the professor's passwords, or maybe even just stealing the laptop to prevent bad grades from being filed. We can exclude situations that are outlandish, like students blowing up the instructor's office. We can probably also exclude other academics from trying to attack the computer, as I would assume that the instructor would freely give course information to other instructors if asked.

Now assume that the database is stored on the instructor's personal laptop, with no network card and no floppy disk drive. Propose at least two security mechanisms that would help counter your threat model (e.g. file or disk encryption, a laptop lock, a safe to store the laptop, a Kevlar laptop sleeve, relocation to Fort Knox...), and analyze the net risk reduction of both. You should justify your estimates for the various incidence rates and costs. While we do want to see numbers for this part, don't worry about figuring out exact costs or risk reductions, guess at some reasonable numbers but don't spend very long at this part of the assignment.

I think the professor should at LEAST put a password lock on the laptop and keep a very close eye on it if they ever remove it from their office. In fact, if students come by for office hours, it should probably be kept in a locked desk. The office should also be locked when the professor is not present, and with a different key than the one for the desk.

For the password lock, it is not very likely that students try to test this security mechanism. The only situation where I can foresee this happening is one in which the professor has the back turned during office hours (because the professor should not be taking this laptop to class). In this case, it would take a very bold student to try to crack the password. Out of all the students who come to office hours (which is a low number to begin with), I estimate 1 in 100 to try testing this security mechanism. And the cost is very low! Most people have passwords on most of their devices and very quickly develop a muscle memory for them. If the professor uses this solution, it will be the single most important preventative measure from students breaking into the database.

For the locked desk, I think that the only situation where this security mechanism is tested is in conjunction with the locked office. It seems unlikely that the professor would not be in the office, and not lock the office door (just because most office doors lock automatically when shut). For this reason, to test this security mechanism, a student would have ALSO had to break into the office. I put the incidence of this happening at 1 in 250. The only reason the incidence is so common, is because instructors will occasionally step out without locking their door, and students may wander in. But to wander in AND THEN try breaking into a locked desk? That situation is unlikely to occur and even less likely to succeed. The cost for this is very low. It may be a minor hassle for the professor to lock and unlock their desk and office, but it's practically the same cost as turning off your faucet before leaving the house and locking the front door, which any rational adult is accustomed to doing. And it would be a very effective security mechanism!

Question 3

Describe how to construct an input that executes an arbitrary command with the privileges of the script. Explain how your input will cause the program to execute your command and suggest how the code could be changed to avoid the problem.

One thing you could do to break this code would be to pass in a command like "username; sudo rm -rf /usr/bin/". Basically, this would pass "username" as the username, and then the shell would go on to insert whatever the attacker put into the program (in this case, deleting stuff). One simple fix would be to just not allow spaces or semicolons in usernames. Users shouldn't be able to set up an account with those in their names, and then before the username gets passed to the script, there should be some statement protections.

Describe how to exploit a race condition to make the function delete the last byte of /what/ever, assuming the program has read and write access to the file (/what/ever) but the user does not. Your description should list what file the fixed string pathname refers to at each important point in the exploit, and why it will work.

The exploit in this situation relies on the window of time where the user inputs a pathname and the system checks whether the user is actually allowed to read/write to that pathname. Basically, while the system program is running the program, a hacker could open a command line prompt and set the symbolic links to symlink("what/ever/", "pathname"), and then before the second stat command, change the symbolic link back to symlink("system/expected/path/", "pathname"). The hacker could then change the symbolic link again to symlink("what/ever/", "pathname"), delete

the last byte, and then before the write command is executed, change the symbolic link back to the expected link. A hacker could do this in one of three ways: 1) by executing the commands at exactly the right time (unlikely), by writing a function to execute the commands at exactly the right time (more likely), or by slowing down the system so much by overloading it that the windows are larger (most likely).