COMP0141 Security
Tutorial 2 Solutions

January 2023

Questions

1. In last week's lectures, you covered the 2019 edition of the design principles of Saltzer and Schroeder. What are the principles? For each question below, consider what you would do, and state which design principle your choice corresponds to. Can you also think of a real world example where this principle has been violated?

    (a) Your respective app store has multiple choices for a given type of app, but they require different permissions. Which do you choose?
    Answer: There's no need to download an app that requires more permissions than it should. This is a straightforward example of the principle of least privilege. Examples of apps violating this involve the somewhat famous flashlight apps (`https://www.theguardian.com/technology/2014/oct/03/android-flashlight-apps-permissions-privacy`).

    (b) Every nuclear missile in your country can be launched from a single machine that one person can login to. Do you give them the code for the door of the room containing the machine, and the doors to the building?
    Answer: Applying the principle of separation of responsibilities would suggest you split up the access to the building, the room and the launch button between at least three people that hopefully are fairly independent. Requiring multiple people to agree on launching nuclear missiles actually happened when Vasily Arkhipov decided to argue against the other two officers on board of the B-59 submarine and stopped the likely start of a nuclear war (`https://en.wikipedia.org/wiki/Vasily_Arkhipov`).

    (c) Some program on your computer asks if it can access some service (e.g., read some file). Do you give it permanent access?
    Answer: The principle of complete mediation would suggest that you avoid giving permanent access to the program. An example failure of this principle is the case of DNS spoofing (i.e., DNS cache poisoning) `https://en.wikipedia.org/wiki/DNS_spoofing`.

    (d) Exams are over and you want to celebrate/drown your sorrows, but you know far too well that you can be forgetful when tired and potentially intoxicated. Do you do anything to account for this?
    Answer: Applying the principle of fail-safe default seems prudent here, and you should only go out if you have set up a plan that deals with predictable issues like losing your keys.

    (e) You've launched a new website that has generated a fair bit of controversy, meaning many people want to see your website, and many would rather it be offline. How do you deal with this?
    Answer: You should of course apply the principle of defence in depth and set up back ups. The important thing is to consider how to have meaningful depth in your defence. Having all your backups in the same place won't help if they all depend on the same network or safety of a single building. If your content isn't very well liked by your host country then having back ups distributed across different countries (that don't have agreements with your host) country seems like another sensible thing to do.

(f) You're trying to build a start up and your security engineer decides he wants to use some secret algorithm he's come up with. He sends you an e-mail describing his scheme in detail and argues that no one will be able to break into your system because his new scheme is amazing. How do you react?

Answer: A good start would be to fire him after explaining the principle of open design. There are likely many people in the world that could try to attack your system, and having the security be evaluated by just your engineer (or team of engineers) will not be as effective as having an open system that can be looked at by anyone. Moreover, internal documentation or e-mails are always prone to being accessed by unintended parties, because someone opens their e-mail on a compromised machine or an engineer leaves the company with a copy of the documentation in hand. Reverse engineering is also a realistic threat.

Failures related to the open design principle including, for example, the content scrambling system that was meant to stop DVD disks being copied (`https://en.wikipedia.org/wiki/Content_Scramble_System#Cryptanalysis`). The enigma machine that the Nazis relied on during WW2 to encrypt messages was also broken in part due to secret information (e.g., manuals, guessing the keyboard was in alphabetical order) obtained by the Allies (`https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma`). In general, Kerckhoffs's principle (from 1883!) states that a cryptosystem's security should not depend on anything but the key being secret. Security through obscurity is also discussed here: `https://www.cs.columbia.edu/~smb/papers/draft-ymbk-obscurity-00.txt`.

(g) You're designing a system that allows users to input information (e.g., login with a password). How would your system react if a wrong value is entered at any point.

Answer: The principle of psychological acceptability would suggest you produce error messages that are understandable by humans. Note that this should also be balanced by security requirements and not revealing too much information. For example, if the user enters a wrong password or a wrong username, it would be better to display an error message that states a wrong username or password has been entered rather than reveal which one was erroneous. This stops an attacker from gaining any new information e.g., knowing that they have the right username or pass word and only need the other.

(h) You're a security consultant and decide to impress your clients by suggesting a combination of many new complicated cryptographic schemes to solve their problems. Should they trust you?

Answer: Absolutely not. The principle of economy of mechanisms applies here. Adding more fluff than necessary does not add any functionalities, but increases the complexity and thus the likelihood of something going wrong. Keep the trusted computing base small.

(i) You need to have an important private conversation with someone. What environment do you choose?

Answer: In any situation that requires a degree of security, applying the principle of prudent paranoia can be helpful. If you have to speak with someone privately, then the sensible thing to do is simply to do so away of anyone or anything that might listen. If a device can record sound and is connected to a network then it can send what it hears over that network. Similarly, your phone also likely includes a camera that can be used to record visual footage, and a GPS that will reveal your location (four spatio temporal points are likely enough to identify you – `https://www.nature.com/articles/srep01376`). Many apps have the permissions required to do this, without forgetting signal intelligence agencies around the world. Anthony van der Meer produced a short film that highlights how much you can learn with remote access to a phone (`https://www.youtube.com/watch?v=NpN9NzO4Mo8`).

More generally, it is good practice to never assume a system is secure without reason, and to remember that a system being secure at a given point in time does not meant it will be secure in the future In the words of Adi Shamir (of RSA fame), "There are no secure systems, only degrees of insecurity." Similarly, if you post something from a private account, there is no guarantee that it will be private. Recent examples are private twitter accounts (`https://www.bbc.co.uk/news/technology-46918859`) and private Facebook pictures, including pictures uploaded to Facebook that were not posted (`https://www.theverge.com/2018/12/14/18140771/facebook-`

`photo-exposure-leak-bug-millions-users-disclosed`). Neither of these were due to attacks. Nonetheless, your level of paranoia should remain healthy, meaning that you should keep it in mind when required but otherwise be able to live. Despite a lot problems, the world more or less functions and is overall better than it used to be.

(j) You've graduated from UCL and work at a big company that handles a lot of data. Your manager comes in to discuss how the data is collected, stored, analysed and discarded. What suggestions do you have for him?
Answer: A good starting point would be to apply the principle of privacy promotion. Not only is this the right thing to do from an ethical point of view, improper handling of private data has caused a lot of damage to many individuals that have had their data collected without their consent, analysed in ways that did not protect their privacy or released in the wild due to a breach. These days, the European General Data Protection Regulation is also in effect and will hopefully mean real consequences when business' are found to mishandle data, with fines of up to 4 of the business' annual worldwide turnover. Google was recently fined €50M by the French CNIL for not providing enough information and transparency regarding advertisers access to personal data (`https://www.theverge.com/2019/1/21/18191591/google-gdpr-fine-50-million-euros-data-consent-cnil`).

(k) Consider the following general code for allowing access to the resource:

```
DWORD dwRet = IsAccessAllowed (...);
 if (dwRet == ERROR_ACCESS_DENIED) {
     // Security check failed
     // Inform user that access is denied.
 } else {
     // Security check OK.
     // Let user use the resource.
 }
```

Answer: This violates the fail-safe default principle. You should base access decisions on permission rather than exclusion, because the set of people that have permission is much smaller than the set of people that do not have permission (i.e., the rest of the world). Try on your own to rewrite the code in a correct way.

2. A company publishes the design of its security software product in a manual that accompanies the executable software. In what ways does this satisfy the principle of open design? In what ways does it not?
Answer: This satisfies the principle open design in the sense that publishing a manual explaining the design allows everyone to understand the functionality that the software is intended to provide. On the other hand, the software itself is not open. If one cannot look at the code, one cannot find bugs in it. This fails to take advantage of the positives that result from the principle of open design.

3. A common technique for inhibiting password guessing is to disable an account after three consecutive failed login attempts. Discuss how this technique might prevent legitimate users from accessing the system. Why is this a design issue?
Answer: Although the aim is to protect against password guessing, this also enables a new attack that affects the availability of the user's account. An adversary can simply attempt to log in with a user's username and an incorrect password so that the user gets blocked. This is possible because the legitimate users and potential adversaries share the same interface i.e., the system does not respect the principle of sharing of responsibilities. Note that this does not mean that systems should be built with users having different authentication interface. It only means that the fact that they share a common interface opens the door to attacks. This is sometimes called the principle of least common mechanisms.

4. When the execution of a piece of software overwrites memory that was not reserved for the program, is it a good idea to revert the execution and reserve more memory? What principle is this solution

violating? What is the good option in this case?
Answer: This is not a good idea because it is very difficult to guarantee that everything in the system will roll back correctly (especially as the complexity of the system increases). This violates the fail-safe default principle. The safe solution is what programs already do: stop and exit with an error to inform the user that something has gone wrong, and to avoid creating more damage in the system.

5. Since you covered a bit about the internet, this exercise is related to giving you a sense of how identifiable you are on the internet.

   (a) A basic privacy question is always "how likely is it that a person is identifiable given some data." One way of measuring this is based on entropy, which essentially translates to the number of bits required to specify some information. Mathematically this is simply expressed as $s = -\log_2 P(x = X)$ for one variable. How much entropy would be required to identify someone? The current world population is around 8,011,000,000, use a calculator (google will do).
   Answer: Answer: $-\log_2(1/8011000000) \approx 33$ bits. The following example is adapted from the one given by the EFF (`https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy`).

   (b) Given that UCL has around 40,000 students and assuming for simplicity that birthdays are uniformly distributed i.e., every day is equally likely to be someone's birthday. How close are you to identifying someone if you know that they study at UCL and their day and month of birth?
   Answer: If we assume your day and month of birth are independent of you going to UCL, we have:
   $s = -\log_2 P(x = X, y = Y) = -\log_2(P(x = X) * P(y = Y)) = -\log_2(x = X) - \log_2(y = Y) = -\log_2(40000/8011000000) - \log_2(1/365) \approx 17.6 + 8.5 \approx 26$ bits,
   which isn't far off being able to identify you (as we need ≈ 33 bits to do so). In general, three or four attributes like this can be enough. Note that we've assumed birthdays are uniformly distributed, which they are not (as are other attributes like postcodes, ...) but this still illustrates the point.

   (c) Based on this, how identifiable do you think you are on the internet? What could be used to identify you? Which design principle should be applied here?
   Answer: Very identifiable, the EFF estimates that just a user agent string like the example you saw in Emiliano's slides ("Chrome London 14.32 1/1/17 128.40.1.76 1280 x 720 Mac OS X English") have around 10 bits of information (`https://www.eff.org/deeplinks/2010/01/tracking-by-user-agent`). Add to that cookies, and other information and it isn't too hard for you to be identified.

   (d) What are the uses of information that can be used to efficiently identify users?
   Answer: Targeted advertising, and other similar forms of targeted content, is one of the main uses of this.

   (e) Do you know of any ways of reducing how identifiable you are?
   Answer: The usual advice is simple: use standard extensions to block ads, trackers, delete cookies, and so on. The GDPR actually had an effect here as you might have noticed since it came into application in 2018. Many websites started serving different pages to users with an European IP address in order to comply with the privacy regulations, some even blocked users. Interestingly, the New York Times stopped serving targeted ads to its European users but still managed to grow its revenue (`https://digiday.com/media/new-york-times-gdpr-cut-off-ad-exchanges-europe-ad-revenue/`).

6. Using the internet involves sending and receiving lots of data. Making sure that this data is secure (think back to confidentiality, integrity and availability) is one problem, another is dealing with the metadata related to these actions.

   (a) Give a brief definition of metadata. Why is the availability of metadata a security problem?
   Answer: Metadata is data that gives information about other data. For example, the data involved in two parties messaging each other is the content of the messages and the metadata

is the time, sender, recipient, ... of the messages. From a security point of view, the issue is that this can still be used in an adversarial way, for example by that two parties communicate even if the content of the communication is not revealed. Moreover, even metadata that does not constitute an identifier by itself can be used, in combination with other metadata, to infer an identity. Metadata falls under the definition of personal data in the GDPR.

(b) Why are anonymous communication tools important? Can you think of examples when they could be needed?

Answer: There are many possible examples, a non-exhaustive list is as follows: incident reporting, whistle blowing, witness protection, freedom of speech, investigation / market research, law enforcement, governments and officials messaging, military organisations, auctions / bidding / stock market, journalists in oppressive countries, etc.

(c) How would you model an adversary in this case? Who is the adversary (i.e., threat)? What is the goal of the adversary? What type of attacks the adversary can perform?

Answer: In this case, the adversary would aim to infer information by looking at traffic, for example to determine who is talking to whom. This could be a third-party adversary, who is just a passive observer, but it could also take the form of malicious nodes that act as an adversary or cooperate with the third-party observer. The possible attacks split into two main categories: passive and active. Passive attacks involve an adversary that simply observes traffic, but does not deviate from the protocol (i.e., it does not interfere with the traffic). Active attacks involve an adversary which, on top of eavesdropping, can also change, drop (or delay) packets, inject his own packets, and son on.

(d) Two popular solutions to anonymous communications are Tor and mix networks. In simple terms, Tor allows a user to encrypt a message several times and send it through the network where each encryption layer is decrypted until it reaches the intended recipient. Mix networks, on the other hand, rely on proxy servers taking in messages from different users and shuffling them ("mixing") before relaying them. What do the different approaches mean in practice?

Answer: Tor provides good anonymity guarantees against a limited adversary that does not control the first and last node of the circuit, but it is vulnerable to an active attacker or a global passive adversary (GPA) that can observe all the traffic. On the other hand, a mix network functions by making traffic analysis hard, making it resistant to active attacks and a global passive adversary. You'll notice that despite Tor appearing to be weaker on paper, it is the only one that is used in practice. This is for two reasons. To start off, there is a debate, with good points on both sides, about how realistic a global passive adversary is. Some argue that a GPA is not currently realistic, while some argue that it is, or that the danger of an adversary that is close enough to being a GPA is realistic. Another important point is that using Tor or mix networks increases the latency of communications. Tor does this to an extent since it involves your traffic going through three nodes, but the problem is much worse for mix networks that have to wait for a certain number of packets to arrive before shuffling them and sending them out. This adds a significant amount of delay, particularly when the number of users is low, which makes it a challenge to get a good amount of users in the first place.

More generally, the number of users is very important, anonymity loves company as the bigger the anonymity set the better protected you are. You can think back to Question 5 and entropy, which is one way of measuring the degree of anonymity in a system. Tor works well because it has by far the most users out of any anonymity system. You can see the number of users, as well as relevant events and other metrics, here: `https://metrics.torproject.org/`.