SIRS Project Topics 2019/20

Important note: the security aspects, both functional and assurance ones, are the essential points where you should focus your work and will be the main metrics to be considered in the grading. In the designed and proposed applications, user interface concerns are secondary.

1. Smartphone as a security token

The smartphone is a digital companion for most people. This work should leverage its *proximity* to another device as part of an increased security solution.

The smartphone can be used for two-factor authentication. For example, the user should have the phone with himself to answer a security challenge posed by a web application. A current solution should reuse existing services such as Duo or similar.

The smartphone can also be used to verify the presence at a specific location, by interacting with other devices. A key can be kept on the phone and then provided to the other devices via a wireless channel (like Wi-Fi, Bluetooth or NFC.) using a secure protocol.
When the phone is present, some resources can be made available to the user. For example, the computer decrypt user files when it senses the phone; when the phone moves away, the directory is encrypted again.

The security concept proposed by this work will have to be implemented in a working prototype, suitable for demonstration of the capabilities.

References:

Duo SDK https://duo.com/product/every-application/supported-applications/apis
KeePass Plugins https://keepass.info/help/v2/plugins.html

2. Ransomware-resistant remote document access

Another focus is the resistance against ransomware attacks. Health care institutions must have backups, with version history, to allow recovery in the case of ransomware attacks that encrypt the data in exchange for money.

Collaborative office applications allow groups of users to create and edit documents remotely. These documents are regularly synchronized between personal devices and the cloud.

In these applications a user, owner of the document, can select contributors to gain access to the document. Documents cannot be accessed by unauthorized parties. If an attacker accesses the servers storing the documents he must not be able to view the documents and if he tries to edit any document, there must be a way to detect the illegal modifications.

In this topic you should design a *cloud-based* solution that allows documents to be shared over a public network in a secure fashion. This application should allow authenticated users to access local and remote files in a transparent way. Data confidentiality must be assured even in the case where an attacker gains physical access to the data storage devices. Illegal modification of the documents by unauthorized users must be detected. The document system should aim to be resistant to *ransomware* attacks.

The system should also provide regular checks of integrity for the stored information, so that the users can be assured that their data is still on the cloud. These proofs should take freshness into account.

References:

For the file synchronization, a tool like Syncthing can be used as a starting point:
https://docs.syncthing.net/

3. Medical Records

Health care institutions gather and store sensitive information from patients with the goal of providing the best care possible. The medical history of a patient is essential to allow correct diagnostic and help the clinical staff act in the shortest time possible. This information is highly sensitive and must be kept private for the responsible staff only. At the same time, the medical records should be accessible by any health care institution to ensure that a patient can be assisted anywhere.

A policy language, like XACML plays a central role here, as it can represent policies with different access control models, like ABAC, RBAC, where different parts of the patient record can be accessed by different people and in different contexts (e.g. regular appointment versus urgent care).

Another aspect is to guarantee data availability. A storage solution will likely rely on external servers, in public cloud providers, accessible through the Internet. This poses a threat, since patient data can be accessed by unauthorized personnel.

In this topic you should define a cloud-based system to store medical records. The records are available for a wide range of users, so the system must provide an interface to manage fine-grained and contextualized access privileges to the records.

References:

XACML https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml#other
XACML library (there are others) https://github.com/att/XACML

4. Automatic Vulnerability Detector

The leader of your group of hackers decided to create a scoring system that rewards the members for any new vulnerabilities that they manage to find. With that goal in mind, he implemented a system that allows members to submit their attacks and gain points.

As a member, you decided to create a program that automatically searches for vulnerabilities in binaries in order to win this intra-group competition!

Your project should provide the following components:

- Compute fingerprinting of binaries through input/output interaction
- Automatically detect common vulnerabilities like calls to "gets" functions or to printf-family functions with user controlled buffers.
- Submit fingerprinting and vulnerabilities to score points.
- Receive and store vulnerabilities from group members
- The leader of the group should be able to see the scoreboard and the exploits of each member.
- Hackers should not be able to see other people's attacks or submit bad attacks for someone else. Basically, try to prevent any kind of cheating.

The primary focus of the system should be the scoring system integrity and information confidentiality.

For the secondary functionality (vulnerability detector) you can choose an existing tool such as the ones below, or implement a rudimentary solution yourself.

References:

AFL - American Fuzzy Lop has been the most widely used fuzzer for a while. It uses genetic fuzzing to find inputs that crashes binaries, among other things.
http://lcamtuf.coredump.cx/afl/

Angr - A state-of-the-art binary analysis engine that allows to symbolically execute a binary and find vulnerabilities and create exploits mostly automatically.
https://angr.io/

BAP - Binary Analysis Platform is a framework for binary analysis. It can be used to detect calls to certain functions for instance.
https://github.com/BinaryAnalysisPlatform/bap

5. Secure child locator

In this scenario, consider the problem of child localization in outdoor or indoor spaces.

Develop a service for smartphone or smartwatch users (e.g., Android) that enables the tracking of children using GPS (e.g. A-GPS), only by their authorized legal guardians (and not by anyone else). As reference, consider the My Ki or the EASYmaxx systems.

You can also design a system for indoors location, where GPS does not work, relying instead on Wi-Fi fingerprinting (e.g. Google Indoors Maps) or Bluetooth Low Energy beacons.

The service to build should consider the secure tracking of the children inside defined geographic fences. There should be a provision for alerts (SOS). Both the children and the responsible adult should be regarded as users of the system, and all stored and communicated data should consider user consent and their privacy.

More advanced solutions should assume limited trust on external servers, for example, the solution may assume the server to be “honest-but-curious”, i.e., it will follow the application protocol but will try to learn as much as possible about its users.

References:

My Ki system https://myki.watch/en/

EASYmaxx
iOS app https://apps.apple.com/us/app/easymaxx-smartwatch/id1375209119
Smartwatch https://www.amazon.de/EASYmaxx-Smartwatch-Armbanduhr-Sprachnachrichten-Standortlokalisierung-blau/dp/B07BZ292D9

Google Maps Indoors https://www.google.com/maps/about/partners/indoormaps/

6. Hardware address tracking

In this scenario, we consider device tracking attacks using their hardware identifiers. Each mobile device has one or more hardware identifiers that are broadcast in local networks, namely the Ethernet and Wi-Fi adapter MAC addresses. The same happens with Bluetooth. A typical system architecture has a data aggregation server (on the cloud), a set of scan servers that send periodic reports of sighted devices, and a query client.

Keeping record of these identifiers and associating them with users can help to recover lost or stolen devices, but it can become a severe privacy violation as we can know of a user's whereabouts and daily routines. One protection for this kind of tracking is having rotating MAC addresses, but with it we lose the ability to legitimately track lost or stolen devices.

The goal of the scenario is to develop a service for smartphones (e.g., Android) or laptops (e.g., Linux) asset management that enables the tracking of devices using hardware identifiers (e.g., MAC addresses) but only by their authorized owners. The system should allow the device owner to know where it is (or where it was last seen), while preventing other users (and attackers) from accessing this same information.

References:

Commercial presentation of the Tile service and its premium version:
https://www.thetileapp.com/en-eu/how-it-works
https://www.thetileapp.com/en-eu/get-premium

Article about a rumored Apple service with similar goals:
https://arstechnica.com/information-technology/2019/06/the-clever-cryptography-behind-apples-find-my-feature/

Paper with a study about MAC address randomization:
https://www.researchgate.net/publication/314361145_A_Study_of_MAC_Address_Randomization_in_Mobile_Devices_and_When_it_Fails

7. Remote Smart car management system

Current automobiles are in fact complex distributed systems, which include different types of critical systems and infotainment systems. Along with this, they also allow to remotely access and even update some of its services, such as update the infotainment and GPS, open the car doors, turn the AC on, or even monitor the levels of gas and tire pressure. However, due to their criticality, some of the systems cannot be accessed such as the brakes or engine functions.

In this work such a system must be developed, which must include:

- In the car: i) an internal critical server, managing the car's critical systems; ii) an internal server used to manage the infotainment systems iii) a server managing non critical systems and external communication.
- In the manufacture side: a service allowing to monitor the car (oil change and other car maintenance features) and the connection of users (via their smartphones) to the car to access some selected features of the car (such as AC, tire pressure, etc…)

One of the required functionalities of the system is having a remote software update procedure with strong code integrity assurance.

You may assume that the car has an always on 4G connection and an interface console. Each part of the system can be emulated/simulated as a virtual machine.