SIRS Project Topics 2018/19

Important note: in the designed and proposed applications, user interface concerns are secondary. The security aspects (such as functional and assurance) are the essential points where you should focus and will be main metrics to be considered in the grading of the work.

Topics available for selection:

1. DDoS Attack Mitigation

DDoS attack that take down servers may be launched via a botnet that enslaves a large number of devices exposed to the Internet and weakly protected devices, like IoT sensors. Attacks can be detected/prevented at the source (client), the destination (server), or in transit (edge). The goal of this project is to design and implement mitigation mechanism for a DDoS attack on a small network. The project consists of three parts:

- Create a (small) botnet. Create a client program to run on infected devices with a self-propagation mechanism that scans public IP addresses for insecure devices and try to access them over the telnet protocol using default login credentials. Light-weight client programs must be implemented to perform the attack (e.g. ping/UDP flooding) from the infected devices. To simulate the DDoS attack you may resort to other existing tools.
- Set-up an Intrusion Detection System at the server. The IDS must be integrated with firewall and router to enable rate-limiting and QoS provisioning. You can use systems such as Snort[*] or Zeek (formerly Bro[†])
- Propose solutions that exploit limited resources on edge devices to detect/prevent attacks in transit.

[*] https://www.snort.org/
[†] https://www.bro.org/zeek.html

2. Monitoring and intrusion recovery of network infrastructures

Network infrastructures rely on secure channels and key management systems to ensure authenticity, integrity and confidentiality of data. When intrusions occur, it is necessary to detect the fault, redeploy the affected servers, regenerate the keys and, if necessary, recover the lost data. A specific example of an attack is a *web site defacement*, where attackers exploit a SQL injection vulnerability and corrupt the data and look of the web site.
To recover, it is necessary to use a monitoring service capable of detecting faults and a recovery service to restore lost data.

The goal of this project is to setup a network infrastructure for a distributed web or mobile application. The distributed application or database should have the required services to allow monitoring and intrusion recovery of network infrastructures. More specifically, the system should provide the following components:

- monitoring service capable of detecting faults
- secure channels established between every component of the system
- key management service that allows unauthorized keys to be revoked and new keys to be generated
- intrusion recovery mechanism capable of restoring lost data and re-establish faulty servers.

We suggest that monitoring, secure channels and key management reuse existing tools, and the recovery mechanism should be the original contribution of your work.

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.

To guarantee data availability, health care institutions in the future might rely on data repositories accessible through the Internet. This poses a threat, since patient data can be accessed by unauthorized personnel. It is also extremely difficult to manage access to data using standard access control mechanisms due to the vast amounts of user, groups and patients and the constant adjustment in privileges that must be done to maintain patient's confidentiality.

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 access privileges to the records. Consider using different access control models, like ABAC, RBAC, and standards, like XACML[‡], that have several open-source implementations.

[‡] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml#other

4. Remote document access

Collaborative office applications allow groups of users to create and edit documents remotely. 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.

5. 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.

One idea is to use the mobile device for two-factor authentication. For example, the user should have the phone with himself to answer a security challenge posed by a web application.

Another idea is to use the phone 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 NFC, Bluetooth, Wi-Fi, etc.) 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.

6. Secure payments using SMS

SMS are still one of the most widely available messaging services. Given this, the goal is to develop an application for a smartphone (e.g., Android or simulated by a small program, sending and receiving SMS-like messages via a UDP channel) that enables the exchange of text messages to transfer money between users.

For this transaction the user must send the identifier of the other user, the amount of money to be transferred, and additional information as needed.

Mechanisms must be added to assure a secure order, i.e., considering integrity, confidentiality and authentication. Support for non-repudiation should also be considered.

Moreover, consider SMS messaging constraints (with a maximum of 120 characters). Techniques such as ciphertext stealing can be used to ensure that these limits are observed.

7. Secure child locator

In this scenario, consider the problem of child localization in outdoor 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 system[§].

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 considered as users of the system, and all stored and communicated data should consider user consent and their privacy.

[§] https://myki.watch/en/