Research report for the Personal Specialization Project

Contents

[Prior knowledge about Cyber Security concepts 2](#_Toc93486576)

[Topics that returned in semester 4 2](#_Toc93486577)

[How this research helped me 10](#_Toc93486578)

[Research into my old code: testing and security 10](#_Toc93486579)

[Previous testing 10](#_Toc93486580)

[The security of my old code 14](#_Toc93486581)

[LoRa research 15](#_Toc93486582)

[What is LoRa? 15](#_Toc93486583)

[Group project implementation 18](#_Toc93486584)

[Resources: 20](#_Toc93486585)

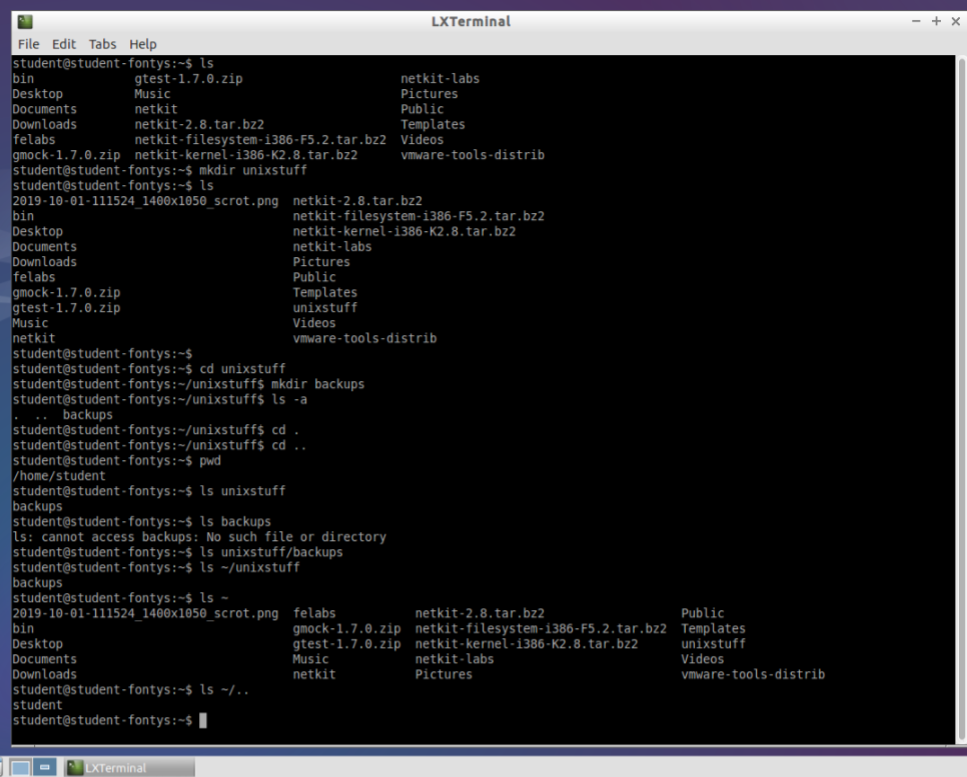
# Prior knowledge about Cyber Security concepts

Coming into this semester I expected most of the information to come from the ICT & Infrastructure field. Therefore, I went back to the Infrastructure Engineering Orienting course from semester 1 on canvas. Given that this was my first and last contact with Infrastructure concepts, it was the sensible step to take. The main purpose was to remember as many concepts as possible before the start of the specialization semester. By doing so, my work throughout the semester has become significantly easier. This semester has been a difficult one for me given that I have only done software engineering at a more advanced level. Re-reading some security concepts has helped me understand the new topics much faster than normal, which means that I would start working on the practical assignments quicker, since the time for theoretical study had reduced.

## Topics that returned in semester 4

* **Operating systems and virtual machines**

This is where I have had the very first contact with a virtual machine. I learned what it is, how it works and why it is useful. Going back through the exercises I have made and the topic learned was helpful because most of my work in semester 4 has been on one or more virtual machines.

Example of an exercise on a virtual machine. I have used basic linux commands for the first time, such as ls, cd, mkdir, or ifconfig.

Text

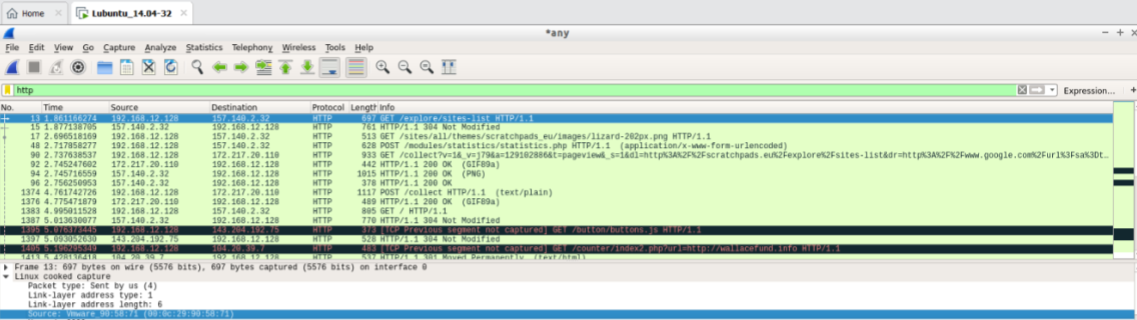
Description automatically generated

* **Introduction to networking**

This has been an introduction to network layers (TCP/IP), along with an explanation about what a network is. It was my first contact with Wireshark, a program that I will extensively use throughout semester 4. I learned the basics about it and because of that I was able to jump straight into the Wireshark exercises without having to stop and study it first. This workshop contained explanations about protocols, information about Linux, and Netkit.

Layered communication. An overview of the TCP/IP protocols:

Overview of the TCP/IP protocols


Intro slides about Netkit
My old Netkit assignment


* **Internet protocol and network devices**

This was a deeper dive into the internet protocols. After a brief explanation of TCP/IP, this workshop talked more in detail about IP, ARP and ICMP. Along the way, I would remember more and more useful Linux commands.Diagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generatedGraphical user interface, text, application

Description automatically generated

* **Transport protocols: TCP and UDP**

Another workshop that proved to be useful, given that I have been working with TCP on my Body of Knowledge documents and with UDP during the second phase of the group project. The slides from that lesson helped me remember what these protocols are and how they work so I can have a better understanding of what I have been requested throughout semester 4.

Diagram

Description automatically generatedA picture containing text

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, application, table

Description automatically generatedText

Description automatically generated

* **HTTP**

This lesson contains information about the Client-Server model. It explains how the HTTP protocol works in detail, containing information about HTTP flow, requests, and responses. JSON is also explained in those slides, another useful topic for not just this semester, but for all that followed.

Chart, waterfall chart

Description automatically generatedA screenshot of a computer

Description automatically generated with low confidence

Text, timeline

Description automatically generated

## How this research helped me

By investing time at the very beginning of the semester, I saved time throughout the rest of the semester. Not having done anything related to ICT & Infrastructure, I did not really remember much of the topics related. By looking at the old slides and exercises that I did, a lot of information came back and that helped me understand the new concepts better and quicker.

# Research into my old code: testing and security

As part of the personal specialization project, I started looking back on my older projects to check if my applications are secure. I also went back to look at unit testing. By making this retrospective, I managed to create a research report about my application security and the way it has evolved.

## Previous testing

* **Testing in semester 2**

Testing in semester 2 came as an optional requirement. Regardless, I chose to implement this feature in my code on a lower scale from both the individual and team assignments. Throughout this semester, I only used unit testing for some of the features of the applications, such as the login or register functions.

* **Testing in semester 3**

As anticipated, testing in semester 3 became a requirement. This time, unit testing will be accompanied by integration testing. I have used these two testing techniques extensively for both the individual and the team assignments. This time, as required, the tests cover the entirety of the code, from login and register functions all the way to testing the classes and the CRUD operations within the applications.

Unit test class example: the post class. The first step is creating a data variable that will be used for testing:Text

Description automatically generated

After this has been achieved, the tests can be written. The test below checks if creating a post is working according to plan:Text

Description automatically generated

After checking the post entirely with all its parameters, they can also be taken one by one. For instance, only the id of the post can be checked:Text

Description automatically generated

Or just the title of the post:Text

Description automatically generated

Example of an integration test. Integration tests checks various modules of the software under development together as a group. This determines whether or not they function together seamlessly as part of the system or whole. In this case, these tests verify if the requests and responses from the HTTP server are working under parameters. For this, a mock data variable needs to be created, just like the unit test.Text

Description automatically generated

The first test creates an account and checks if the HTTP response is the right one (200). The next two create wrong data on purpose to check if the HTTP server will react in the right way and reject the request.Text

Description automatically generatedText

Description automatically generated

## The security of my old code

The focal point of this research was to review my old code when it comes to its security. I was pleased to find out that my old websites were resistant to attacks such as SQL injection, cross-site scripting, cross-site request forgery or path traversal. However, there was no 2-factor authentication, and the token could be relatively easily intercepted by an attacker.

This is an early implementation of user authentication. If the (logged in) user does not have the privileges necessary to access a specific web page, if they enter the URL of that page or access it by pushing a button on the website, they will be redirected to the home page and asked to log in with a different account, with more privileges.Text

Description automatically generated

This is an easy fix for SQL injection attacks. Before the form data is sent to the SQL server, the system checks the user input, thus making an SQL injection attack impossible.

Protection has been made against XSS and CSRF attacks as well. In the early versions on my web applications, there were no tokens or sessions. The account information was sent via the URL (the username and the password). This was extremely insecure. After more progress with the application, I changed the way of data transmission first through sessions and then through tokens. No data will be sent via URL and if the attacker tries to access a file they are not supposed to, they will be instantly redirected to the home page.

Throughout the projects, I noticed that they became more complex and more secure. The newer the project, the more security features and tests it would contain. It was an useful exercise, because I have also experimented with web hacking on my older applications, instead of just using the DVWA machine.

# LoRa research

After the conclusion of phase 1 of the project, me and my teammates have received a task for phase 2, completely unrelated to our work prior to that and to anything we have learned. More than two thirds in the semester we found ourselves having to work with something we have not even heard of: a LoRa device. Naturally, we started doing research about it in order to be able to work with it.

## What is LoRa?

LoRa (short for long range) is a spread spectrum modulation technique derived from chirp spread spectrum (CSS) technology. LoRa is a long range, low power wireless platform that has become the main wireless platform of Internet of Things. Designed for IoT communications, LoRa devices enable the connection between remote end nodes and low power wide area networks for delivery to analytics applications. LoRa is the physical silicon layer, or wireless modulation, used to create the long-range communication link. Transceivers configured with LoRa devices are embedded into end nodes, or sensor devices, designed for a multitude of industry applications. Sensors capture and transmit data to gateways over distances near and far, indoor, and outdoor, with minimal power requirement. Gateways send information via Wi-Fi, Ethernet or Cellular to the network server, which is responsible for network management functions like over-the-air activation, data de-duplication, dynamic frame routing, adaptive rate control, traffic management, and administration. Applications interpret the data collected by sensors, applying techniques like machine learning and artificial intelligence to solve business problems for a smarter planet.

Possible attacks/vulnerabilities:

• Replay Attack - The ABP-activated end devices are using static keys which are preprogrammed into the device. Therefore, after resetting, an ABP-activated end device will reuse the frame counter value from 0 with the same keys. In this case, an attacker can grab messages in the last session with larger counter values and reuse it in the current session. Besides resetting, another method to restart the counter is a counter overflow. After the counter value reaches its maximum value, the counter will be reset and will restart from 0. With counter values from the last session and the same session keys, an attacker can also replay previous messages to cut off the communication between the end device and the server. This holds both for ABP and OTAA. However, attacking an ABP-activated end device will take less time as both reset and overflow work if the attacker has the ability to reset end devices.

There are a few things to do in order to avoid attacks:

* Minimize the use of ABP, and use OTAA if possible If insist on using ABP:
* Adopt new keys periodically
* Protect end devices physically, i.e., use non-volatile memory to preserve the counter value and avoid sudden change
* Rekey every time the counter reaches its maximum value

• Eavesdropping - The attack is designed to compromise the encryption method of LoRaWAN. By sniffing the wireless traffic between the gateway and the end device, the attacker can use the corresponding relationship between 2 messages with the same counter value to decrypt the ciphertext. After the attack, the attacker can compromise the confidentiality of the system, and obtain sensor data transmitted in the system. If LoRaWAN is used to transmit secret data, this attack can cause serious privacy issues.

There are a few things to do in order to avoid attacks:

* Replace the counter value by a nonce, which is generated from a cryptographically secure pseudo random number generator
* Rekey on reset (easier to achieve by OTAA as well)

• Bit-flipping Attack - The integrity between application server and network server is not checked. Uplink messages are encrypted then signed. After they are received by the network server, the network server will use NwkSKey to check the signature of the message. After this, encrypted messages are accepted in the network server and then handled to application server. Between the network server and the application server, data can be modified during the handling, because when messages arrive in the application server, the integrity of ciphertext will not be checked anymore.

There are a few things to do in order to avoid attacks:

* Run the integrity check value at the application server instead of the network server
* Repurpose protocol fields: replace the CRC by a MIC

• ACK Spoofing - In most of the cases, the gateway is internet-facing, making the LoRaWAN system more vulnerable. Also, building a malicious gateway is feasible. Through attacks such as UDP spoofing, a malicious gateway can be added into a LoRaWAN system. A protocol flaw is that the ACK for uplink message doesn’t indicate which message it actually confirmed, it only confirms the last message it receives. So, it is possible that the malicious or hacked gateway can keep the confirmation and use it for future messages.

There are a few things to do in order to avoid attacks:

* Recall in Bit-flipping Attack Mitigation part that we use MIC to guarantee integrity throughout the whole transmission.
* Apply MIC to both the connection to the network server and the application server

• LoRa Class B Attacks - The vulnerability of LoRa class B networks is that the beacons are not encrypted. Since there is no encryption, all the information that the beacon contains is in plaintext. If there is any crucial data transmitted, the attacker is able to read it. In addition, though it is claimed that CRC is used to protect the integrity of the beacon’s common part (Time and NetID), CRC depends on physical layer parameters, and it can also be calculated by the attacker. If the attackers have the basic knowledge of BCNPayload, the attackers can build and send their own malicious beacon with malicious parameters, and these beacons will be received and processed by the end devices.

Counter measure:

* change the PHY CRC to a MIC → authenticating the beacon frames

## Group project implementation

Group project task: a LoRa sensor network communicates sensor data via a gateway to a cloud environment, to be used in applications. What are the options, alternatives, and challenges to monitor the security of such an IoT application?

The task of the group was to develop further on that research question and create a proof of concept.

Work was done using Node-Red, a platform on which we have also created the user interface. Node-red is a platform made to simplify building apps for low-end devices.  
By the usage of connecting nodes in an order, each performing a single step in the process of the application with pre-build and optimized code for low-end devices, node-red allows for the developers to write efficient code more easily for low-end devices. Since the app will be running on a raspberry pi 3B performing as gateway as well, the project team decided to not build a full-stack application with many features. Node-red helped the project team to keep the application simple and efficient, prioritizing the raspberry pi 3B on being an efficient gateway instead of a host for the application.Diagram

Description automatically generated

Code snippet from a template found online. This code has been designed for the first interactions with a LoRa device. This code helped me, and my teammates better understand how a LoRa device operates. We have simulated long range communication by having a team member go outside the building and try to send and receive packets.Text

Description automatically generated with low confidence

For the hardware work we started off by erasing any data from our devices that was left over from previous projects, like our SD card. Then, we simply tried to understand the LoRa protocol and see if our sending devices were working by simply connecting them and seeing if they could receive and send (full duplex). The next step was to connect a sending device to our gateway. To do this, we first uploaded our filesystem to the SD card. We went over each pin of our raspberry pi and shield to see if any of them were broken or malfunctioning. Then, we connected everything together. We then tested if we could send data from our sending device through our gateway to the Things Network. Once that worked we started to connect our sending device through the gateway to our own application. We then used hackRF to see if we could intercept any useful data. However, we weren’t able to, but learned more about the protocol in the end and got a better grasp of the vulnerabilities of our setup.

# Resources:

* <https://www.semtech.com/lora/what-is-lora>
* <https://git.fhict.nl/I415849/lorawan-monitoring-project>
* <https://fhict.instructure.com/courses/8896/modules>
* My personal code
* <https://custom-writing.org/blog/cyber-security-topics-for-research>
* <https://usessaywriters.com/blog/cyber-security-research-topics.py>
* <https://resources.workable.com/cyber-security-policy>
* <https://amtrustfinancial.com/blog/insurance-products/what-to-do-after-a-data-breach-or-cyber-attack>