

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος **Secure Internet of Things Applications**

Τίτλος Ασκησης	Τελική Εργασία 2024 - B' μέρος
Όνομα φοιτητή – Αρ. Μητρώου (όλων σε περίπτωση ομαδικής εργασίας)	Ανδριανόπουλος Βασίλειος ΜΠΚΕΔ2303
	Συμεωνίδης Ιωάννης ΜΠΚΕΔ2239
	Αθανασάκης Ιωάννης ΜΠΚΕΔ2301
Ημερομηνία παράδοσης	25 Μαρτίου 2024



Εκφόνηση της άσκησης

The project involves the development of an IoT system responsible to perform human comfort measurement and movement detection. The intelligence/complexity of the IoT system is open and will be the responsibility of each team. The basic functionality will need to measure human comfort and at least 4 different human movement patterns furthermore it should detect falls and alert an authority (e.g. a hospital) to assist each person in danger.

The STM32 MCU will be used to collect and encrypt the measurements from the various available sensors. The data stored in the MCU and the data transmitted to the hub (emulated by matlab), must be encrypted using the AES library provided by the instructors. The Matlab implementation of the AES which was provided by the instructors will be used to decrypt the data in Matlab and plot them.

Then the security methodology presented during the lectures will have to be applied

to the designed IoT system. Assets and threats should be provided and explained.

Identify at least 5 security mechanisms and explain their protection capabilities.

Analyze the vulnerabilities of the IoT application you have designed and draw a diagram showing them.



1. Εισαγωγή

1.1 Background

Στο συνεχώς εξελισσόμενο τοπίο του Διαδικτύου των Πραγμάτων (IoT), η ανάγκη παρακολούθησης περιβαλλοντικών παραμέτρων αλλά και της ανίχνευσης των ανθρώπινων δραστηριοτήτων έχει γίνει ολοένα και πιο σημαντική για μια ποικιλία εφαρμογών, συμπεριλαμβανομένων της υγειονομικής περίθαλψης. Η ενσωμάτωση της τεχνολογίας των αισθητήρων, της κρυπτογράφησης και της ανάλυσης δεδομένων σε πραγματικό χρόνο παίζει καθοριστικό ρόλο, παρέχοντας αυξημένη ασφάλεια για την υγεία των ανθρώπων.

Αυτό το project αξιοποιεί το STM32L475 IoT Discovery Kit για την καταγραφή περιβαλλοντικών δεδομένων και ανθρώπινης δραστηριότητας, χρησιμοποιώντας Advanced Encryption Standard (AES) για ασφαλή μετάδοση δεδομένων και MATLAB για ανάλυση και οπτικοποίηση των δεδομένων.

1.2 Στόχος

Ο πρωταρχικός στόχος αυτού του project είναι η ανάπτυξη ενός συστήματος που θα εκτελεί τα παρακάτω:

- Ακριβής παρακολούθηση των περιβαλλοντικών συνθηκών, ειδικά της θερμοκρασίας και της υγρασίας, σε πραγματικό χρόνο.
- Ανίχνευση ανθρώπινων δραστηριοτήτων όπως ανάπαυση, περπάτημα και τρέξιμο, καθώς και ανίχνευση πτώσης, για τη βελτίωση της παρακολούθησης της υγείας και των συστημάτων απόκρισης έκτακτης ανάγκης.
- Διασφάλιση της ασφάλειας (μας άρεσε σαν όρος 😊) των δεδομένων που μεταδίδονται μέσω κρυπτογράφησης AES. (tbd)
- Παροχή φιλικής προς το χρήστη οπτικοποίηση των δεδομένων, διευκολύνοντας την άμεση και ενημερωμένη λήψη αποφάσεων με βάση τα δεδομένα της θερμοκρασίας και της υγρασίας.



2. Επισκόπηση συστήματος

Αυτή η ενότητα, περιγράφει την αρχιτεκτονική του συστήματος περιβαλλοντικής παρακολούθησης και ανίχνευσης δραστηριότητας, αναφέροντας λεπτομερώς τα κύρια στοιχεία και τη διαδικασία ροής των δεδομένων.

2.1 Αρχιτεκτονική

STM32L475 IoT Discovery Kit

Λειτουργεί ως η κεντρική μονάδα, εξοπλισμένη με διάφορους αισθητήρες για την ανίχνευση θερμοκρασίας, υγρασίας και κίνησης (επιταχυνσιόμετρο). Διαθέτει επίσης LED για την ένδειξη της κατάστασης του συστήματος.

Αλγόριθμος κρυπτογράφησης AES

Η μονάδα είναι ικανή να κρυπτογραφεί τα δεδομένα του αισθητήρα σε πραγματικό χρόνο, διασφαλίζοντας ότι όλα τα δεδομένα που μεταδίδονται για ανάλυση είναι ασφαλή από υποκλοπή ή παραβίαση. (tbd)

Εφαρμογή MATLAB

Λαμβάνει τα κρυπτογραφημένα δεδομένα, τα αποκρυπτογραφεί και επεξεργάζεται τις μετρήσεις του αισθητήρα. Περιλαμβάνει αλγόριθμους για ανίχνευση δραστηριότητας και πτώσης που βασίζονται σε δεδομένα του επιταχυνσιόμετρου, καθώς και λειτουργίες για απεικόνιση της θερμοκρασίας και της υγρασίας.

2.2 Ροή Δεδομένων

Λήψη δεδομένων: Το σύστημα διαβάζει περιοδικά τις τιμές από τους αισθητήρες θερμοκρασίας, υγρασίας και επιταχυνσιόμετρου στο IoT device.

Μετατροπή δεδομένων: Οι μετρήσεις του αισθητήρα μετατρέπονται σε συμβολοσειρές της μορφής '(x,y,z,temp,hum)' και προετοιμάζονται για κρυπτογράφηση.

Κρυπτογράφηση δεδομένων: Οι παραπάνω συμβολοσειρές κρυπτογραφούνται χρησιμοποιώντας τον αλγόριθμο κρυπτογράφησης AES.

Μετάδοση: Τα κρυπτογραφημένα δεδομένα του αισθητήρα αποστέλλονται μέσω σειριακής σύνδεσης σε υπολογιστή.



Αποκρυπτογράφηση και Επεξεργασία των δεδομένων: Με την παραλαβή, η εφαρμογή MATLAB αποκρυπτογραφεί τα δεδομένα χρησιμοποιώντας τον αντίστοιχο αλγόριθμο αποκρυπτογράφησης AES. Στη συνέχεια, τα αποκρυπτογραφημένα δεδομένα αναλύονται για την εξαγωγή μετρήσεων θερμοκρασίας, υγρασίας και επιταχυνσιόμετρου.

Ανίχνευση δραστηριότητας: Η εφαρμογή αναλύει τα δεδομένα του επιταχυνσιόμετρου για να ανιχνεύσει διαφορετικές ανθρώπινες δραστηριότητες (ανάπαυση, περπάτημα, τρέξιμο) και πιθανές πτώσεις που μπορεί να έχει το άτομο.

Για την ανίχνευση των δραστηριοτήτων παίρνουμε τις απόλυτες τιμές του array (x,y,z) και συγκεκριμένα το μέτρο του διανύσματος (x,y,z).

Σύμφωνα με πολλές μετρήσεις που πραγματοποιήσαμε, το μέτρο των δεδομένων που επιστρέφει η IoT συσκευή όταν βρίσκεται σε ακινησία, είναι μεταξύ του 1000 και του 1025.

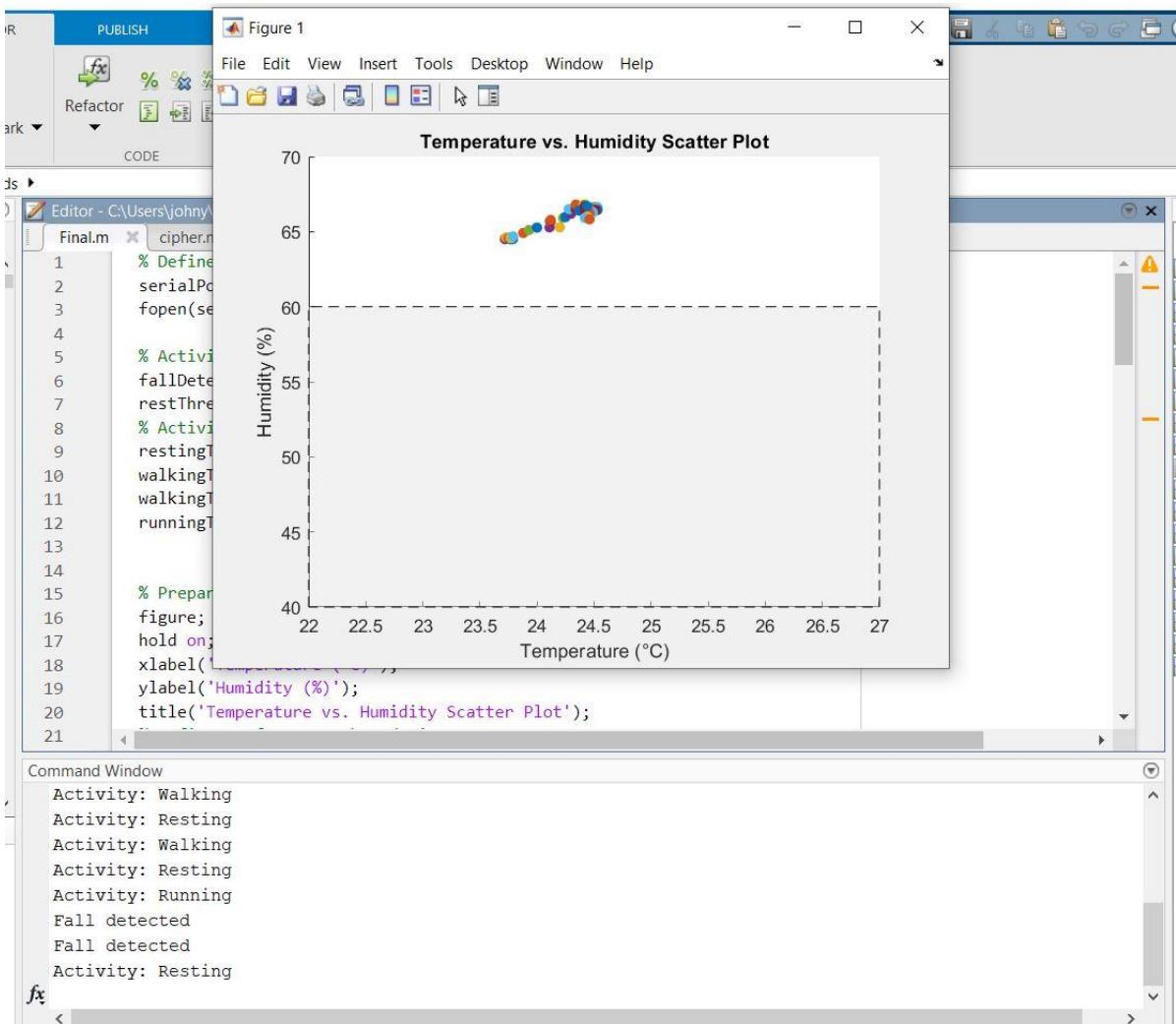
Όταν περπατάει, είναι μεταξύ του 1026 και του 1400 ενώ όταν τρέχει είναι πάνω από 1400.

Για την ανίχνευση του πεσίματος, αρκεί να έχει πιάσει μια επιτάχυνση πάνω από 2000 και ξαφνικά να πάει κοντά στο 1000 (ακινησία)

Οπτικοποίηση: Τα δεδομένα θερμοκρασίας και υγρασίας οπτικοποιούνται σε πραγματικό χρόνο μέσω διαγραμμάτων διασποράς, εμφανίζοντας τις τρέχουσες περιβαλλοντικές συνθήκες.

Συγκεκριμένα, αυτά τα σημεία θα πρέπει να βρίσκονται μέσα στο διάστημα [22,27] και [40,60] για την θερμοκρασία και την υγρασία αντίστοιχα για να είναι εντός comfort zone.

Παρακάτω στην απεικόνιση, βλέπουμε πως τη συγκεκριμένη χρονική στιγμή το άτομο δε βρισκόταν σε comfort zone.



2.3 Προχωρημένη Ροή Δεδομένων

Κατά την απεικόνιση των δεδομένων δραστηριότητας, παρατηρήσαμε πως το IoT δεν είναι πλήρως ακριβές.

Για παράδειγμα, σε περιπτώσεις ακινησίας, το μέτρο του διανύσματος (x,y,z) από το επιταχυνσόμετρο, δεν είναι σταθερό.

Αυτό σημαίνει πως το output μπορεί να αποκλίνει από resting σε walking.

Επομένως, εφαρμόσαμε την εξής πολιτική.

Θα διαβάζει τα 3 statuses και θα εμφανίζει το επικρατέστερο.

Πχ. Αν το output στη Matlab ήταν το παρακάτω,



Resting
Resting
Walking

Θα επέστρεφε το Resting καθώς είναι το επικρατέστερο.

Επίσης, display της δραστηριότητας θα γίνεται μόνο την πρώτη φορά που θα εμφανίζει το status και κάθε επόμενη αλλαγή στο status της δραστηριότητας. Με αυτόν τον τρόπο, θα πετύχουμε ένα καθαρό display με βάση το current activity.

Επίσης, αν εντοπίσει πέσιμο, το κάνει εκτυπώνει κατευθείαν.

3. Υλοποίηση

Η ενότητα της υλοποίησης αφορά τα τεχνικά βήματα που έγιναν για την υλοποίηση του project στο Keil αλλά και την ανάπτυξη λογισμικού λήψης και επεξεργασίας δεδομένων στο MATLAB.

3.1 Ρύθμιση Υλικού

```
// Include sensor drivers from BSP
#include "stm32l475e_iot01_tsensor.h"
#include "stm32l475e_iot01_hsensor.h"
#include "stm32l475e_iot01_accelero.h"
#include "mbed.h"

// Maximum number of elements the application buffer can contain
#define MAXIMUM_BUFFER_SIZE 32

// Create a DigitalOutput object to toggle an LED.
static DigitalOut led(LED1);

// Create a BufferedSerial object with a default baud rate.
static BufferedSerial serial_port(USBTX, USBRX);

int main(void) {
    // Initialize sensors
    BSP_TSSENSOR_Init();
    BSP_HSENSOR_Init();
    BSP_ACCELERO_Init();

    // Set desired properties for serial communication (9600-8-N-1).
```



```
serial_port.set_baud(9600);
serial_port.set_format(
    /* bits */ 8,
    /* parity */ BufferedSerial::None,
    /* stop bit */ 1);

// Sensor read values
float temperature = 0;
float humidity = 0;
int16_t pDataXYZ[3] = {0}; // Accelerometer data

while (1) {
    // Get sensor values
    temperature = BSP_TSENSOR_ReadTemp();
    humidity = BSP_HSENSOR_ReadHumidity();
    BSP_ACCELERO_AccGetXYZ(pDataXYZ);

    // Temperature and humidity processing
    int tmpInt = static_cast<int>(temperature);
    int tmpFrac = static_cast<int>((temperature - tmpInt) * 100);
    int humInt = static_cast<int>(humidity);
    int humFrac = static_cast<int>((humidity - humInt) * 100);

    // Prepare buffer for sending data
    char buf[MAXIMUM_BUFFER_SIZE];
    int length = snprintf(buf, sizeof(buf), "(%d,%d,%d,%d.%d,%d.%d)\n",
        pDataXYZ[0], pDataXYZ[1], pDataXYZ[2],
        tmpInt, tmpFrac, humInt, humFrac);

    // Write the formatted data to the serial port
    serial_port.write(buf, length);

    ThisThread::sleep_for(100ms); // Add a delay for readability and to prevent flooding
}
}
```

Drivers αισθητήρων: Το πρόγραμμα ξεκινά συμπεριλαμβάνοντας drivers για τους αισθητήρες Θερμοκρασίας (tsensor), υγρασίας (hsensor) και επιταχυνσιόμετρου (accelero) από το Board Support Package (BSP) για τον κόμβο IoT STM32L475.

Αυτά τα προγράμματα οδήγησης επιτρέπουν στο πρόγραμμα να επικοινωνεί με τους αισθητήρες στην πλακέτα.

mbed Framework: Η συμπερίληψη του mbed.h υποδηλώνει ότι το πρόγραμμα χρησιμοποιεί το πλαίσιο mbed, το οποίο παρέχει API για τον έλεγχο των χαρακτηριστικών υλικού μιας



ευρείας σειράς μικροελεγκτών ARM Cortex-M, κάνοντας την ανάπτυξη ταχύτερη και ευκολότερη.

Μεταβλητές

Μέγιστο μέγεθος buffer: Καθορίζει μια σταθερά σε bytes για το μέγιστο μέγεθος μιας προσωρινής μνήμης δεδομένων, η οποία χρησιμοποιείται για τη διατήρηση των δεδομένων πριν από την αποστολή τους μέσω της σειριακής θύρας.

DigitalOutput: Χρησιμοποιείται για να υποδείξει οπτικά την κατάσταση της συσκευής ή την ολοκλήρωση ορισμένων λειτουργιών μέσω του LED.

Κύριο Πρόγραμμα

Αρχικοποίηση: Ο αισθητήρας θερμοκρασίας, υγρασίας και το επιταχυνσιόμετρο αρχικοποιούνται για να ξεκινήσει η μέτρηση των δεδομένων.

Ανάγνωση και επεξεργασία αισθητήρα: Μέσα σε έναν βρόχο που δε σταματάει ποτέ, το πρόγραμμα διαβάζει δεδομένα από τον αισθητήρα θερμοκρασίας, τον αισθητήρα υγρασίας και το επιταχυνσιόμετρο. Στη συνέχεια επεξεργάζεται αυτά τα δεδομένα, μετατρέποντας τις ενδείξεις θερμοκρασίας και υγρασίας σε ακέραιους.

Μορφοποίηση και μετάδοση δεδομένων: Τα δεδομένα του αισθητήρα μορφοποιούνται σε string και στέλνονται στη σειριακή θύρα, επιτρέποντάς τους να διαβάζονται από το πρόγραμμα στη MATLAB.

Αναμονή: Προστίθεται μια μικρή καθυστέρηση στο τέλος κάθε επανάληψης του βρόχου για τον έλεγχο του ρυθμού συλλογής και μετάδοσης των δεδομένων.

3.2 Ρύθμιση Λογισμικού

```
% Define Serial Port
serialPort = serial('COM4', 'BaudRate', 9600, 'Terminator', 'LF', 'Timeout', 10);
fopen(serialPort);

% Activity and Fall Detection Parameters
fallDetectionThreshold = 2000;
restThreshold = 1000;
% Activity thresholds
restingThresholdUpper = 1025; % Upper limit for resting
walkingThresholdLower = 1026; % Lower limit for walking
walkingThresholdUpper = 1400; % Upper limit for walking
runningThresholdLower = 1401; % Lower limit for running
```



```
% Prepare the figure for plotting Temperature and Humidity
figure;
hold on;
xlabel('Temperature (°C)');
ylabel('Humidity (%)');
title('Temperature vs. Humidity Scatter Plot');
% Define comfort zone boundaries
comfortZoneTemp = [22, 27];
comfortZoneHum = [40, 60];
% Plot the comfort zone area
patch([comfortZoneTemp(1), comfortZoneTemp(1), comfortZoneTemp(2),
comfortZoneTemp(2)], ...
[comfortZoneHum(1), comfortZoneHum(2), comfortZoneHum(2), comfortZoneHum(1)], ...
[0.9, 0.9, 0.9], 'LineStyle', '--', 'FaceAlpha', 0.5);

% Initialize variables
activityBuffer = {"", "", ""};
lastDisplayedActivity = "";
fallDetected = false;

try
    while true
        % Check if data is available
        if serialPort.BytesAvailable > 0
            dataLine = fgetl(serialPort); % Read a line of data
            % Parse the received data
            C = textscan(dataLine, '(%f,%f,%f,%f,%f)');

            % Extract accelerometer data
            x = C{1};
            y = C{2};
            z = C{3};
            % Calculate the magnitude of the acceleration vector
            g = sqrt(x.^2 + y.^2 + z.^2);

            % Fall detection logic
            if g > fallDetectionThreshold && ~fallDetected
                % Potential fall detected
                fallDetected = true;
            elseif g < restThreshold && fallDetected
                % Fall confirmed
                disp('Fall detected');
            end
        end
    end
end
```



```
fallDetected = false; % Reset fall detection
end

% Activity detection logic using thresholds
currentActivity = "";
if g <= restingThresholdUpper
    currentActivity = "Resting";
elseif g >= walkingThresholdLower && g <= walkingThresholdUpper
    currentActivity = "Walking";
elseif g >= runningThresholdLower
    currentActivity = "Running";
end

if ~isempty(currentActivity)
    activityBuffer = [activityBuffer(2:end), currentActivity];
    if all(strcmp(activityBuffer{1}, activityBuffer)) && ~isempty(activityBuffer{1})
        currentConsensusActivity = activityBuffer{1};
        if ~strcmp(currentConsensusActivity, lastDisplayedActivity) &&
~isempty(currentConsensusActivity)
            disp(['Activity: ' currentConsensusActivity]);
            lastDisplayedActivity = currentConsensusActivity;
        end
    end
end

% Extract and plot temperature and humidity
temp = C{4}; % Combine integer and fractional parts
hum = C{5}; % Combine integer and fractional parts
scatter(temp, hum, 'filled'); % Plot current reading
drawnow;

% Print values to MATLAB Command Window (optional)
%fprintf('X: %.2f, Y: %.2f, Z: %.2f, Temp: %.2f °C, Hum: %.2f%%\n', x, y, z, temp, hum);
end
pause(0.1); % Adjust the pause to control the reading frequency
end
catch e
    disp(['Error: ', e.message]);
    fclose(serialPort);
end
```



Ρύθμιση επικοινωνίας

Αρχικοποίηση σειριακής θύρας: Το script ξεκινά δημιουργώντας μία μεταβλητή σειριακής θύρας serialPort για την επικοινωνία με το IoT στο COM4.

Άνοιγμα σειριακής σύνδεσης: Η εντολή fopen ανοίγει τη σύνδεση, επιτρέποντας τη μετάδοση των δεδομένων.

Παράμετροι ανίχνευσης πτώσης και δραστηριότητας: Το fallDetectionThreshold χρησιμοποιείται για την ανίχνευση πιθανών πτώσεων με βάση την επιτάχυνση, ενώ το restThreshold βοηθά στην ανίχνευση ξεκούρασης (αλλά και την επιβεβαίωση της πτώσης)

Ορίζουμε και τις πρόσθετες μεταβλητές που θα οριοθετούν τα επιπλέον επίπεδα δραστηριότητας (περπάτημα, τρέξιμο) με βάση την επιτάχυνση.

Ρύθμιση οπτικοποίησης δεδομένων

Προετοιμασία σχήματος: Γίνεται η δημιουργία του σχήματος για την απεικόνιση των δεδομένων θερμοκρασίας και υγρασίας.

Περιλαμβάνει τη σκίαση της comfort zone περιοχής.

Δραστηριότητα και λογική για την ανίχνευση πτώσης

Κύριος βρόχος: Μέσα σε έναν βρόχο που δε θα σταματάει ποτέ, ελέγχουμε συνεχώς για διαθέσιμα δεδομένα. Όταν τα δεδομένα είναι διαθέσιμα, τα διαβάζει, και αναλύει τις τιμές της επιτάχυνσης, θερμοκρασίας και υγρασίας και προχωρά με τον αλγόριθμο ανίχνευσης πτώσης και ταξινόμησης δραστηριότητας.

4 Security Analysis

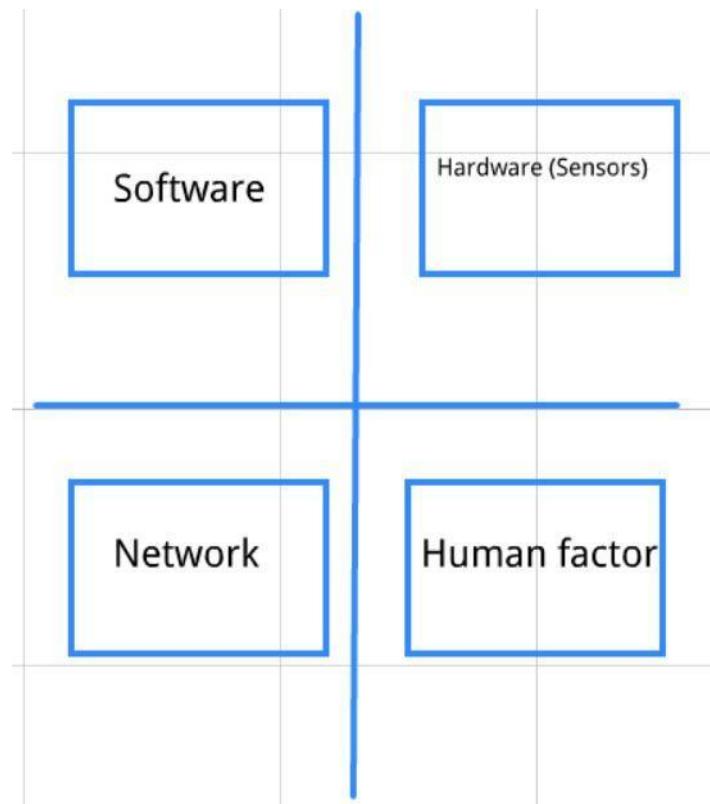
Επιθέσεις πλευρικού καναλιού έναντι αισθητήρων

Στην παρούσα ενότητα θα παρουσιαστούν συνοπτικά και εν παραλλήλω ορισμένες εκ των σημαντικότερων απειλών που μπορεί να αναφύονται έναντι της ιατρικής εφαρμογής ομού μετά των στρατηγικών αντιμετώπισης τους. Καθώς η εφαρμογή βασίζεται εν πολλοίσι στην πλακέτα STM32 και τις εφαρμογές που επικοινωνούν με αυτή (π.χ. λήψη δεδομένων ή αποστολή εντολών προς τους αισθητήρες) η ανάλυση των απειλών θα βασιστεί στα τρία κύρια χαρακτηριστικά που συναποτελούν την εφαρμογή ήτοι: σχεδιασμός πλακέτας (hardware), ανάλυση με βάσει τους αισθητήρες, και τέλος ανάλυση επί τι βάσει του λογισμικού (software).



Επί την βάσει αυτού του τρίπτυχου (ήτοι σχεδιασμός, αισθητήρες, εφαρμογές) μπορούν να προσδιοριστούν πέντε βασικά είδη απειλών, και κατόπιν και τα αντίμετρα τους, ως ακολούθως:

- Οι επιθέσεις πλευρικού καναλιού (side channel attacks) έναντι αισθητήρων,
- Οι επιθέσεις πλευρικού καναλιού επί του επιπέδου εξυπηρετητή (π.χ. περιβάλλον Cloud),
- Η απειλή των κινητών τμημάτων κώδικα (mobile code),
- Οι επιθέσεις άρνησης υπηρεσιών σε επίπεδο δικτύου (DdoS),
- Το ανθρώπινο δυναμικό ως δυνάμει απειλή (human factor).



Σχηματική απεικόνιση των τεσσάρων (4) βασικών αδυναμιών ασφαλείας εν σχέσει με την ιατρική εφαρμογή



Οι παραπάνω απειλές, έστω και ονομαστικά, μπορούν να καταδείξουν μια σειρά από αδυναμίες που άπτονται του υλικού και του λογισμικού και που αναπόφευκτα αποτελούν τμήμα της υλοποίησης του κώδικα επί της πλακέτας. Οι αδυναμίες που κάνουν εφικτή την ύπαρξη των παραπάνω απειλών, είναι, ανάμεσα σε άλλες, και οι ακόλουθες:

- **εργοστασιακές παράμετροι ως προς την κατασκευή της πλακέτας (hardware):** υφίστανται δύο βασικά ζητήματα σε αυτό το σημείο. Αφενός, επικρατεί ετερογένεια ανάμεσα στους κατασκευαστές πλακετών, κι έτσι δυσχεραίνετε η ομοιογενείς προσέγγιση στην λήψη αντίμετρων κατά των απειλών. Αφετέρου, η επιφάνεια της πλακέτας είναι συχνά σε αντίστροφη αναλογία προς τα στοιχεία που βρίσκονται εντός της. Η μικρή απόσταση των στοιχείων της πλακέτας επιτρέπει στους επιτιθέμενους να χρησιμοποιήσουν συνδυαστικά είτε τις εκροές είτε και τα ετερογενή στοιχεία για να διενεργήσουν επιθέσεις με στόχο την απόσπαση πληροφοριών .
- **Αδυναμίες που προκύπτουν από τις ίδια την φύση των IoT συσκευών (π.χ. στον κλάδο της υγείας, healthcare):** ανάμεσα στις βασικές αδυναμίες των IoT συσκευών γενικά περιλαμβάνονται η ετερογένεια στον κλάδο των κατασκευαστών, η έλλειψη κοινών προτύπων, οι περιορισμοί αναφορικά με την ισχύ και τους πόρους που θα πρέπει να ξοδευτούν για την ενίσχυση της άμυνας τέτοιων συσκευών, και επίσης το γεγονός της συνεχούς και αδιάλειπτης λειτουργίας μιας τέτοιας συσκευής άνευ επιτηρήσεως (plug & forget), όπως και η συχνή απουσία διεπαφών φιλικών προς τον χρήστη. Όλα τα παραπάνω στοιχεία αποκρυσταλλώνονται σε δύο βασικές παραμέτρους, αφενός το πλήθος των συσκευών (που συνεχώς αυξάνεται), αφετέρου η διεύρυνση της αλυσίδας επιθέσεων (attack chain) που συνεχίζει ν' αυξάνεται (hopping) στον βαθμό που αυξάνουν οι συσκεύες IoT που επιτρέπουν την προσέγγιση του τελικού στόχου. Όλες αυτές οι πτυχές κάνουν ώστε οι επιτιθέμενοι να προχωρούν στην στόχευση τέτοιων συσκευών όταν επιθυμούν να επιτεθούν σε κρίσιμες υποδομές όπως τα νοσοκομεία .
- **Αδυναμίες των εκάστοτε στρατηγικών κυβερνοασφάλειας (cybersecurity strategies):** Πρόκειται για την αντίστροφη οπτική σε σχέση με τις δύο προηγούμενες περιπτώσεις. Η προβληματική εδώ εστιάζει στην σαφή απουσία περιεκτικών χαρτογραφήσεων, καθώς οι επιμέρους κατηγοριοποιήσεις απειλών αφενός περιλαμβάνουν πληθώρα στοιχείων και αφετέρου έχουν ως αφετηρία τους πολλά διαφορετικά στοιχεία που οδηγούν σε διαφορετικές κατατάξεις απειλών (π.χ. κάποιες κατηγοριοποιήσεις εκκινούν από τις συσκευές στόχους, ενώ άλλες από το είδος της κρίσιμης υποδομής που δυνάμει δέχεται την επίθεση κλπ.). Υπό αυτό το πρίσμα οι αναλύσεις και κατηγοριοποιήσεις είτε αφήνουν εκτός στοιχεία άλλων χαρτογραφήσεων είτε προχωρούν σε μια επικάλυψη από διαφορετική οπτική .



Φυσικό επίπεδο (Physical level/sensors)

Το βασικό χαρακτηριστικό της ανά χείρας πλακέτας είναι η ενσωμάτωση πολλαπλών αισθητήρων (π.χ. κίνησης, θερμοκρασίας, μέτρησης υγρασίας κλπ.) εντός ενός μικρού σχετικά χώρου (λόγω και του μεγέθους των πλακετών εμπορίου). Δεδομένοις της πολλαπλής αυτής ενσωμάτωσης υφίστανται αντίστοιχα πολλαπλές εκροές (π.χ. ενέργεια, φωτεινότητα, θερμοκρασία, εντοπισμός κίνησης κλπ.) που οι επιτιθέμενοι δύνανται ποικιλοτρόπιας να εκμεταλλευτούν για να αποσπάσουν πληροφορίες. Χαρακτηριστικά παραδείγματα τέτοιων εκροών αποτελούν οι επιθέσεις πλευρικού καναλιού, ορισμένες εκ των οποίων αναλύονται (με βάσει τους αισθητήρες της πλακέτας) αμέσως κατωτέρω:

- **Επιθέσεις ανάλυσης ισχύος:** Καθώς η εν λόγω πλακέτα φέρει έναν ψηφιακό μετατροπέα (Time-to-digital Converter, TDC) ένας επιτιθέμενος θα μπορούσε να προβεί σε μετρήσεις ηλεκτρικής ισχύος και ανάλογα με τις τάσεις που εμφανίζουν οι τελευταίες (υψηλή-χαμηλή) να μπορέσει να παραβιάσει τον αλγόριθμο κρυπτογράφησης και να αποκτήσει πρόσβαση σε αντίστοιχα σημαντικές πληροφορίες. Οι επιθέσεις ανάλυσης ισχύος είναι ένας παραδοσιακός τρόπος διενέργειας επιθέσεων κατά (και) lightweight συσκευών που μπορούν να συμβούν είτε με εγγύτητα στον στόχο είτε απομακρυσμένα. Χαρακτηριστική περίπτωση είναι εκείνη όπου ο επιτιθέμενος στέλνει plaintexts προς κρυπτογράφηση ενώ παράλληλα μετρά τον χρόνο επεξεργασίας από πλευράς CPU (συνήθως είναι μια template attack).
- **Επιθέσεις εισαγωγής σφάλματος (Fault Injection SCAs):** Εδώ ο επιτιθέμενος πρέπει να έχει επαφή με την συσκευή IoT και ο στόχος του είναι να εισάγει σφάλματα επί των διάφορων τμημάτων hardware ώστε να διαταράξει την ομαλή λειτουργίας τους. Χαρακτηριστικό παράδειγμα αποτελεί η clock glitch SCA όπου ο επιτιθέμενος γνωρίζοντας ότι οι πλακέτες διατηρούν δεδομένα στην σειριακή εκτέλεση (latch data που διατηρούνται ανάλογα με το rising edge του ρολογιού και την χαμηλή, falling, ή υψηλή, rising, τάση του τελευταίου), επιχειρεί είτε μέσω του φωτός είτε μέσω της αλλαγής στην τάση ισχύος να διαταράξει την ομαλή λειτουργία της συσκευής (μέσω του input που δίνει) και έτσι να προκαλέσει, αν όχι την καταστροφή της, τουλάχιστον την μη αναμενόμενη λειτουργία αυτής (arbitrary execution) και κατά τον τρόπο αυτό να λάβει γνώση πληροφορίας ή να προσπελάσει μηχανισμούς αυθεντικοποίησης χρήστη (authentication mechanisms).
- **Βασική στρατηγική αποφυγής τέτοιων επιθέσεων είναι η απομόνωση των συσκευών IoT καθώς και το decoupling μεταξύ ισχύος και αποτελεσμάτων που αυτή δίνει .**

Επίπεδο λογισμικού (κινητά τμήματα κώδικα/mobile code)

Καθώς η ανά χείρας ιατρική εφαρμογή προϋποθέτει την συγγραφή γραμμών κώδικα (scripts), όμοια με έτερες ιατρικές εφαρμογές, πρέπει να ληφθούν υπόψη και οι ανάλογες απειλές ασφαλείας που προκύπτουν από την εκτέλεση των εκάστοτε γραμμών. Εκ των βασικών κενών ασφαλείας κατά την χρήση κινητού κώδικα από διαδικτυακά αποθετήρια, αποτελούν αφενός η χρήση τμημάτων legacy κώδικα που δεν μπορούν να αναβαθμιστούν καταλλήλως (π.χ. depreciated methods/functions κλπ.) και αφετέρου η εκτέλεση των εν λόγω τμημάτων κατά μη ντετερμινιστικό τρόπο που οδηγεί σε απώλεια πληροφορίας προς όφελος του επιτιθέμενου ή σε δυσλειτουργία της εφαρμογής (arbitrary code execution). Οι πλέον πιθανές απειλές που μπορούν να προκύψουν από κινητά τμήματα κώδικα, μεταξύ άλλων, είναι οι ακόλουθες :



- Η δυνατότητα του επιτιθέμενου να αποκτήσει πληροφορίες για την τρέχουσα εφαρμογή (inspection),
- Η δυνατότητα του επιτιθέμενου να προχωρήσει σε τροποποίηση του όλου script (modification/arbitrary code execution),
- Η δυνατότητα του επιτιθέμενου να προχωρήσει σε επίθεση τύπου Replay, εκτελώντας απομακρυσμένα την εντολή και κατόπιν ανταλλάσσοντας πληροφορίες με τον καθαυτό χρήστη (Replay),
- Η δυνατότητα του επιτιθέμενου να προχωρήσει σε άρνηση υπηρεσιών, με το να μην εκτελεστεί ορθά η εντολή του κινητού κώδικα και άρα να επηρεαστεί η συνολική λειτουργία της εφαρμογής μέχρι του σημείου να μην είναι πλέον εκτελέσιμη (Denial of Service).
- Βασική στρατηγική ασφαλείας στο σημείο αυτό είναι η εκπαίδευση του προσωπικού μετά της κατάλληλης απομόνωσης των κινητών τμημάτων κώδικα (sandboxing), και όπου αυτό είναι εφικτό η αποφυγή χρήσης πεπαλαιωμένων γλωσσών που δεν συμβαδίζουν με τις απαιτήσεις ασφαλείας ή που οι software developers μιας υγειονομικής μονάδος δεν μπορούν να υποστηρίξουν/αναβαθμίσουν κατάλληλα.

Επίπεδο εξυπηρετητή (Cloud)

Από την άλλη μεριά οι επιθέσεις πλευρικού καναλιού σε περιβάλλον Cloud δεν απαιτούν άμεση πρόσβαση στην συσκευή, και αντιθέτως επικεντρώνουν περισσότερο στην απομακρυσμένη δυνατότητα υφαρπαγής δεδομένων, όταν τα τελευταία αποθηκεύονται αφού έχει γίνει λήψη τους από τους αισθητήρες της συσκευής. Ένας βασικός τρόπος επίθεσης κατά της αποθήκευσης των δεδομένων από τους αισθητήρες σε περιβάλλον Cloud (κάλλιστα οι νοσοκομειακές μονάδες μπορούν να το χρησιμοποιήσουν διότι η ροή δεδομένων από αισθητήρες είναι εύλογα αυξανόμενη σε 24ωρη βάση εντός των κλινικών κλπ.) είναι η παρακολούθηση της δραστηριότητας της cache memory από πλευράς του δράστη, ιδίως σε περιπτώσεις όπου ο επιτιθέμενος και οι νόμιμοι χρήστες διαμοιράζονται το ίδιο περιβάλλον (π.χ. όταν ο επιτιθέμενος είναι insider). Ενδεικτικές υποπεριπτώσεις τέτοιων επιθέσεων είναι και οι ακόλουθες :

- Prime & Probe: Ο επιτιθέμενος προβαίνει σε χρονομέτρηση κατά την είσοδο γραμμών μνήμης σε τμήματα της cache memory, εν συνεχείᾳ επανεισάγει τις ίδιες γραμμές μνήμης και ανάλογα με το αν αυτές φορτώνουν με μεγαλύτερη ή μικρότερη βραδύτητα μπορεί να εντοπίσει αν ο νόμιμος χρήστης έχει χρησιμοποιήσει τα ίδια τμήματα μνήμης με εκείνον και άρα να αποκτήσει γνώση για το περιεχόμενο των γραμμών .
- Evict & Time: Εδώ ο επιτιθέμενος μπορεί να αποσπάσει το κλειδί για την αποκρυπτογράφηση (αν έχει προτέρα γνώση της χαρτογράφησης των τμημάτων μνήμης στον λόγω Cloud) με το να στέλνει plaintexts προς κρυπτογράφηση. Εν συνεχείᾳ προβαίνει σε φόρτωση και έξωση (evict) των lookup tables (πινάκων αληθείας) και όταν στείλει το αίτημα για κρυπτογράφηση (ciphertext) μπορεί να μετρήσει τον χρόνο που απαιτείται για την εκτέλεση της τελευταίας, αν το χρονικό διάστημα είναι σχετικά μικρό τότε το look-up table έχει διατηρηθεί στην μνήμη ενώ σε αντίστροφη περίπτωση έχει γίνει evict .



- Βασική στρατηγική αντιμετώπισης είναι, μεταξύ άλλων, ο ντετερμινιστικός χρόνος εκτέλεσης των ενεργειών σε περιβάλλον Cloud (deterministic time of execution), ή η εν γένει δυνατότητα διάρρηξης ανάμεσα στην εκτέλεση της ενέργεια και στην δυνατότητα χρονομέτρησης (time padding) καθώς και η εφαρμογή απομόνωσης των χρηστών σε περιβάλλον Cloud (sandboxing).

Επίπεδο Human Factor (Ανθρώπινου παράγοντος)

Η απειλή που προέρχεται από το ίδιο το προσωπικό (είτε δρουν ως εκ των ένδον απειλές είτε λόγω πλημμελούς προετοιμασίας) αφορά, κατά κύριο λόγο, σε δύο βασικά σημεία. Αφενός, στην πλημμελή εκπαίδευση του προσωπικού αναφορικά με ζητήματα που άπτονται της κυβερνοασφάλειας και της πρόληψης κατά των σχετιζόμενων απειλών. Αφετέρου, αφορά στην αδυναμία των IoT συσκευών σε ένα ιατρικό περιβάλλον να ενσωματώσουν στοιχεία φιλικά προς τον χρήστη κατά την αξιοποίηση της εκάστοτε ιατρικής εφαρμογής ή του εκάστοτε ιατρικού εξοπλισμού. Βασική στρατηγική αντιμετώπισης αυτών των κενών ασφαλείας είναι εν μέρει η εκπαίδευση του προσωπικού και εν μέρει η συμμόρφωση των νοσοκομειακών μονάδων με τα εκάστοτε πρότυπα ασφαλείας των διεθνών και ευρωπαϊκών οργανισμών (π.χ. ENISA, NIST κλπ.).

Επίπεδο Δικτύου (Network level)

Η τελευταία απειλή που θα μελετηθεί άπτεται της άρνησης υπηρεσιών (DDoS) κατά την μεταφορά των δεδομένων και των συναγερμών από τους αισθητήρες στην διεπαφή του ιατρικού προσωπικού. Η βασική λογική της άρνησης υπηρεσιών είναι η αύξηση του traffic εντός του νοσοκομειακού δικτύου μέχρι του σημείου όπου προκαλείται ανάσχεση στην παροχή των προβλεπόμενων υπηρεσιών (π.χ. αδυναμία αποστολής δεδομένων από τον αισθητήρα στο περιβάλλον της διεπαφής του ιατρικού προσωπικού).

Οι επιθέσεις άρνησης υπηρεσιών εδράζονται στο επίπεδο του δικτύου (network level) και παρουσιάζουν μια σειρά από εφικτές παραλλαγές κατά την διενέργεια τους έναντι των κρίσιμων υποδομών της υγειονομικής περίθαλψης. Καθώς ο αριθμός των συσκευών και των διασυνδέσεων τους βαίνει αυξανόμενος, είναι όλο και πιο εφικτό ένας επιτιθέμενος να αξιοποιήσει μια DDoS εκμεταλλευόμενος την όλο και μεγαλύτερη διασύνδεση ανάμεσα στις επιμέρους κλινικές του νοσοκομείου αλλά και την αντίστοιχη διασύνδεση ανάμεσα σε διαφορετικές Κρίσιμες Υποδομές κλπ.

Στον βαθμό που τα πρωτόκολλα επικοινωνίας ανάμεσα στις IoT συσκευές δεν ενέχουν τον ίδιο βαθμό ασφαλείας όπως σε έτερες συσκευές (εν μέρει διότι οι επεξεργαστική ισχύ των πρώτων είναι μικρότερη εν συγκρίσει με εκείνη των δεύτερων), ο επιτιθέμενος συνήθως επιλέγει είτε να στείλει μια συνεχώς αυξανόμενη ροή δεδομένων προς το πρωτόκολλο ICMP (Internet Control Message Protocol) είτε προς εκείνο του UDP (User Datagram Protocol) με στόχο να προκαλέσει καθυστερήσεις ή και ολική διακοπή επικοινωνίας ανάμεσα στους αισθητήρες και



την ιατρική διεπαφή που λαμβάνει τα δεδομένα και τους συναγερμούς (π.χ. fault detection alerts κλπ).

Παρεμφερείς περιπτώσεις επιθέσεων στο επίπεδο δικτύου με εμπλοκή κάποιου ενδιάμεσου (Man in the Middle Attacks), αποτελούν και οι επιθέσεις Replay & False data Injection Attacks. Στην μεν πρώτη περίπτωση ο ενδιάμεσος στέλνει δεδομένα που υπό κανονικές συνθήκες αποστέλλουν οι αισθητήρες και έτσι αρχίζει μια συναλλαγή ανάμεσα στους νόμιμους χρήστες και στον επιτιθέμενο που επιτρέπει στον τελευταίο ν αποσπάσει πληροφορίες. Στην δε δεύτερη περίπτωση, συμβαίνει το αντίστροφο και ο επιτιθέμενος στέλνει μη αποδεκτά δεδομένα (false data) για να προκαλέσει εν τέλει αστοχίες στις διαδικασίες των αισθητήρων (invalid operations). Μια τυπική στρατηγική έναντι τέτοιων επιθέσεων στο επίπεδο δικτύου αποτελεί κυρίως η συχνή διαδικασία ελέγχου του traffic του δικτύου για τον εντοπισμό ασυνήθιστων συμπεριφορών στην κυκλοφορία κατόπιν σύγκρισης με τον προηγουμένως καταγεγραμμένο ρυθμό κυκλοφορίας εντός του δικτύου (π.χ. behavior-based detection κλπ.).

Βιβλιογραφία

Staddon, E., & Losci, V., & Mitton, N.(2021). Attack categorization for IoT Applications in critical infrastructures, a survey. *Applied sciences*, 11, 7228, 1-39.

Shurman, M., & Khrais, R.M., & Yateem,A.A.(2020). IoT denial of service attack detection and prevention using hybrid IDS. Paper presented at the Conference: 2019 International Arab Conference on Information Technology (ACIT), 3. December, 2019.

Nifakos, S., & Chandramouli, K., & Nikolaou, C.K., & Papachristou, P., & Koch, S., & Panaousis, E., & Bonacina, S., (2021). Influence of human factors on cybersecurity within healthcare organisations: a systematic review. *Sensors*, 21, 5119, 1-25, 13 & 18.

Bazm, M.M., & Lacoste, M., & Sudholt, M., & Menaud, J.M.(2017). Side Channels in the Cloud: Isolation Challenges Attacks, and Countermeasures. *Computer Science,Engineering*,1-14.

Jansen, W.A., & Winograd, T., & Scarfone, K.(2008). Guidelines on active content and mobile code Recommendations of the National Institute of Standards and technology. NIST: U.S. Department of Commerce.

Hertzbleed: Turning power side-channel attacks into remote time attacks on x86.1-14.

Gangolli, A., & Mahmoud, Q.H., & Azim, A.(2022). A systematic review of fault injection attacks on IoT systems. *Electronics*, 11, 1-24.