

Modulo IoT

Guida pratica al modulo IoT Security con EXPLIoT e DIVA

Ambito autorizzato e sicurezza

Le procedure vanno eseguite soltanto sulla DIVA, sullo smartwatch vulnerabile fornito con il kit, sulle catture incluse o su sistemi per i quali si dispone di autorizzazione esplicita.

Regole essenziali:

1. non usare la modalità continuità del multimetro su una scheda alimentata;
2. non collegare tra loro le linee VCC del target e dell'adattatore EXPLIoT Nano;
3. collegare sempre una massa comune prima dei segnali logici;
4. verificare il livello logico, normalmente 3,3 V nel banco descritto;
5. eseguire un backup prima di scrivere flash o registri;
6. conservare separatamente file originali e file modificati;
7. evitare scritture casuali su hardware non sacrificabile.

Struttura del percorso

Modulo	Natura prevalente
UART	identificazione e comunicazione seriale
JTAG/SWD	identificazione, dump, patching e periferiche
I ² C	ricognizione, sniffing e dump EEPROM
SPI	ricognizione, sniffing e analisi firmware
BLE su DIVA	ricognizione, GATT e dimostrazione vulnerabile
BLE su smartwatch	acquisizione, analisi e controllo GATT
802.15.4/ZigBee	preparazione, scoperta, cattura, replay e decrittazione

Le tre attività ZigBee presentate nel manuale per la configurazione con due DIVA non sono tre nuovi esercizi: sono varianti dei laboratori già descritti con una DIVA.

Fondamenti necessari

Lettura del datasheet e identificazione del package

Prima di cercare un bus sulla scheda occorre identificare il componente e il package corretto. La stessa famiglia di microcontrollori può essere venduta con package diversi e quindi con numerazioni dei pin differenti. Nel target DIVA descritto dal manuale il microcontrollore STM32WB55 usa il package VFQFPN68.

Procedura generale:

1. leggere la marcatura del componente;
2. confrontare forma e numero dei pin con il datasheet;
3. localizzare il riferimento del pin 1;
4. consultare la tabella delle funzioni alternate;
5. distinguere il numero fisico del pin dal nome logico del segnale;
6. verificare la continuità verso test point o header.

Continuità, tensione e attività digitale

La continuità risponde alla domanda: “questi due punti sono collegati elettricamente?”. Va misurata a scheda spenta.

La tensione risponde alla domanda: “quale livello assume questo pin rispetto a GND?”. Va misurata con il circuito alimentato, evitando cortocircuiti con la punta della sonda.

Indicazioni tipiche:

- GND: continuità verso la massa;
- VCC: tensione stabile, nel banco circa 3,3 V;
- TX UART inattivo: spesso alto, ma variabile durante la trasmissione;
- RX UART: può apparire alto o basso e spesso si identifica per esclusione;
- clock e dati di I²C/SPI: richiedono logic analyzer o oscilloscopio per essere riconosciuti con affidabilità.

UART

UART è asincrona: non esiste una linea di clock condivisa. Trasmettitore e ricevitore devono usare lo stesso baud rate e la stessa configurazione dei frame.

Collegamento del kit:

```
1 Target TX  -> EXPLIoT Nano RX, pin 3
2 Target RX  -> EXPLIoT Nano TX, pin 5
3 Target GND -> EXPLIoT Nano GND, pin 1
4 VCC        -> non collegare
```

TX e RX sono incrociati. Caratteri illeggibili indicano spesso baud rate errato, massa mancante o livelli logici incompatibili.

JTAG e SWD

JTAG e SWD sono interfacce di debug. Possono consentire arresto del core, lettura e scrittura della memoria, accesso ai registri e programmazione della flash, salvo protezioni attive.

Nel percorso vengono usati:

- il datasheet per la mappatura dei pin;
- JTAGenum per la scansione automatica delle combinazioni;
- EXPLIoT Nano come adattatore;
- OpenOCD come server di debug;
- Telnet come console di comando di OpenOCD.

Memoria flash e I/O memory-mapped

La flash contiene il firmware persistente. Prima di leggerla occorrono indirizzo iniziale e dimensione, ricavati dal datasheet e dalla memory map.

Nei microcontrollori ARM Cortex-M le periferiche sono spesso memory-mapped: registri GPIO e periferiche occupano indirizzi nello stesso spazio della memoria. Leggere due dump in stati diversi e confrontarli permette di individuare i byte correlati a LED o altri segnali.

I²C

I²C usa SDA e SCL, normalmente con linee open-drain e resistenze di pull-up. Ogni periferica risponde a un indirizzo. Le EEPROM aggiungono indirizzi interni di parola e specifiche sequenze di lettura e scrittura.

Per analizzare una transazione occorre distinguere:

- START e STOP;
- indirizzo slave e bit read/write;
- ACK/NACK;
- indirizzo interno;
- dati trasferiti.

SPI

SPI usa normalmente CS, CLK, MOSI e MISO. Non esiste un indirizzo slave sul bus: il dispositivo è selezionato dalla linea CS. Il decoder deve usare correttamente polarità e fase del clock, indicate come CPOL e CPHA.

Una memoria SPI si controlla tramite opcode. Il datasheet indica comandi di identificazione, lettura, programmazione, cancellazione e registri di stato.

BLE e GATT

BLE separa il trasporto radio dal modello applicativo GATT. I concetti chiave sono:

- indirizzo BLE del dispositivo;
- servizio, identificato da UUID;
- caratteristica, identificata da UUID e handle;
- operazioni read, write e notification;
- pairing e chiavi di sessione o a lungo termine.

L'handle è un identificatore locale della tabella ATT. Il suo significato si ricava osservando UUID, permessi e traffico generato dall'applicazione legittima.

802.15.4 e ZigBee

IEEE 802.15.4 definisce il livello radio e MAC usato da ZigBee. I parametri utili nel laboratorio sono canale, PAN ID, indirizzi e frame catturati.

Il percorso distingue:

- scoperta della rete;
- cattura passiva;
- filtraggio dei frame del proprio banco;
- replay controllato;
- analisi della procedura di associazione e delle chiavi.

Modulo 1 - UART

Unità 1.1 - Identificazione UART mediante datasheet e continuità

Tipo: laboratorio pratico.

Obiettivo

Identificare TX e RX partendo dai pin USART del microcontrollore e verificando la connessione verso gli header della DIVA.

Procedura ragionata

1. Spegnerla la DIVA.
2. Identificare il package VFQFPN68 e il riferimento del pin 1.
3. Cercare nel datasheet i pin USART_TX e USART_RX disponibili.
4. Numerare gli header candidati sulla scheda.
5. Misurare la continuità tra ciascun pin del microcontrollore e gli header.
6. Registrare una tabella pin fisico, funzione e header corrispondente.

Risultato atteso

Una mappa ripetibile dei pin UART. Il semplice beep del multimetro dimostra la connessione elettrica, non che la periferica UART sia già abilitata dal firmware.

Errori comuni

- package o orientamento del chip errati;
- scheda alimentata durante la continuità;
- confusione tra più istanze USART;
- contatto instabile su pin molto piccoli.

Unità 1.2 - Identificazione UART mediante misure elettriche

Tipo: laboratorio pratico.

Obiettivo

Individuare GND, VCC, TX e RX quando il datasheet o il pinout non sono disponibili.

Sequenza consigliata

1. trovare GND a scheda spenta mediante continuità;
2. alimentare la scheda e cercare un livello stabile compatibile con VCC;
3. osservare i pin durante boot o generazione di log per trovare TX;
4. identificare RX per esclusione e confermarlo con una prova di comunicazione;
5. non usare VCC nel collegamento UART con EXPLIoT Nano.

Il TX può sembrare una tensione quasi costante quando non trasmette. Il momento migliore per osservarlo è l'avvio, quando molti dispositivi emettono log.

Unità 1.3 - Comunicazione seriale con il target

Tipo: laboratorio pratico.

Obiettivo

Collegare EXPLIoT Nano, determinare il baud rate e verificare se l'interfaccia fornisce soli log oppure una console interattiva.

Collegamento

Usare massa comune, TX incrociato con RX e nessun collegamento VCC.

Ricerca del baud rate

Il manuale propone il plugin di scansione del framework EXPLIoT:

```
1 ef> run uart.generic.baudscan -p /dev/ttyUSB0
```

Il plugin prova baud rate comuni e stima la percentuale di caratteri ASCII. Il valore con testo coerente è il candidato migliore; nel banco descritto viene mostrato 9600 baud.

Verifica finale

```
1 screen /dev/ttyUSB0 9600
```

Classificare ciò che compare:

- log di boot soltanto;
- prompt personalizzato;
- shell interattiva;
- richiesta di autenticazione;
- nessuna uscita, che può indicare pin o baud rate errati.

Modulo 2 - JTAG e SWD

Unità 2.1 - Identificazione mediante datasheet

Tipo: laboratorio pratico.

La procedura ripete il metodo UART, cercando però JTCK, JTMS, JTDO e JTDI. Il risultato deve essere una mappa verificata con continuità. Lo stesso principio è applicabile a SWDIO e SWCLK quando il target usa SWD.

Unità 2.2 - Scansione automatica con JTAGenum

Tipo: laboratorio guidato.

Arduino Nano, già predisposto o programmato con `jtagenum.hex`, viene collegato ai pin sospetti. L'ordine iniziale non è importante perché il software prova combinazioni.

```
1 screen /dev/ttyUSB0 115200
```

Nel terminale si avvia la scansione con il comando previsto da JTAGenum. Più risultati `FOUND` possono differire soltanto per TRST, segnale opzionale non incluso nella scansione descritta.

Consegna

Riportare la corrispondenza tra pin Arduino e segnali JTAG, spiegando come è stata confermata.

Unità 2.3 - Estrazione del firmware

Tipo: laboratorio pratico.

Obiettivo

Usare OpenOCD per collegarsi al core e leggere la regione flash.

Flusso

1. identificare e cablare i pin;
2. alimentare la DIVA solo dopo avere verificato i collegamenti;
3. avviare OpenOCD con i file di configurazione forniti;
4. collegarsi alla console Telnet;
5. ricavare base e dimensione della flash dal datasheet;
6. arrestare il core, eseguire il dump e poi riprenderne l'esecuzione.

Il manuale usa una base flash pari a `0x08000000` e mostra un comando `dump_image` specifico per la DIVA. La dimensione va sempre verificata prima di applicare la procedura ad altri microcontrollori.

Validazione del dump

Controllare:

- dimensione del file;
- presenza di dati non uniformi;
- stringhe leggibili;
- vettore iniziale plausibile;
- hash del file originale.

Unità 2.4 - Patching e riscrittura del firmware

Tipo: laboratorio avanzato.

Il manuale propone di trovare una password hardcoded nel dump, modificarla con un editor esadecimale e riscrivere l'immagine nella flash tramite OpenOCD.

Procedura disciplinata:

1. duplicare il dump originale;
2. cercare la stringa e confermare che appartenga alla challenge corretta;
3. sostituire con una sequenza della stessa lunghezza, salvo conoscenza del formato;
4. conservare hash e offset della modifica;
5. usare `reset init`, scrittura con cancellazione e `reset` come indicato dal manuale;
6. verificare il comportamento dalla console DIVA.

Questo è un laboratorio di integrità del firmware: la vulnerabilità è la presenza di un segreto modificabile e non protetto, non l'editor esadecimale in sé.

Unità 2.5 - Accesso alle periferiche memory-mapped

Tipo: laboratorio avanzato.

Obiettivo

Correlare lo stato dei LED ai registri GPIO.

Metodo differenziale

1. identificare il GPIO coinvolto e la sua base nella memory map;
2. acquisire un dump dei registri nello stato A;
3. cambiare lo stato dei LED mediante la challenge;
4. acquisire un dump nello stato B;
5. confrontare i file;
6. calcolare `indirizzo base + offset modificato`;
7. sperimentare scritture controllate sul solo registro individuato.

Il confronto tra due stati riduce il numero di registri candidati e rende l'analisi più robusta rispetto alla scrittura casuale.

Modulo 3 - I²C

Unità 3.1 - Ricognizione della EEPROM

Tipo: esercizio di analisi preparatoria.

Usare la marcatura della EEPROM e il datasheet [24LC256.pdf](#) per compilare una scheda con:

- capacità e organizzazione;
- tensione di alimentazione;
- pin SDA e SCL;
- pin di selezione indirizzo;
- indirizzo slave risultante;
- dimensione dell'indirizzo interno;
- page size;
- sequenze di current address read, random read e page write.

Il manuale richiama il prefisso binario `1010` tipico di questa famiglia e il bit finale read/write. La configurazione dei pin di indirizzo completa l'indirizzo effettivo.

Unità 3.2 - Sniffing I²C e autenticazione

Tipo: laboratorio pratico.

Obiettivo

Acquisire la comunicazione tra microcontrollore ed EEPROM durante la challenge di autenticazione.

Procedura

1. identificare SDA, SCL e GND;
2. collegare il logic analyzer senza pilotare le linee;
3. avviare la challenge dalla console DIVA;
4. acquisire la transazione;
5. configurare il decoder I²C;
6. individuare indirizzo slave, indirizzo interno e dati;
7. confrontare i dati letti con la password immessa.

Nel materiale didattico compare la stringa `admin`, che evidenzia il rischio di memorizzare o confrontare credenziali in chiaro su una EEPROM osservabile.

Unità 3.3 - Dump della EEPROM I²C

Tipo: laboratorio pratico.

EXPLIoT Nano viene collegato alle linee I²C secondo le specifiche fornite. Il manuale indica di ponticellare i pin necessari del Nano e usare il plugin:

```
1 ef> run i2c.generic.readEEPROM -c 24AA256 -w dump_i2c.bin
```

Controlli

- verificare modello e capacità;
- salvare il dump senza sovrascriverlo;
- calcolare un hash;
- cercare stringhe e strutture coerenti;
- confrontare offset interessanti con la cattura del laboratorio precedente.

Modulo 4 - SPI

Unità 4.1 - Ricognizione della memoria SPI

Tipo: esercizio di analisi preparatoria.

Dal datasheet della memoria ricavare:

- capacità;
- pin CS, CLK, MOSI e MISO;
- tensione e frequenza massima;
- opcode di identificazione e lettura;
- registri di stato e protezione;
- modalità SPI supportate.

L'analisi del datasheet è necessaria per evitare di interpretare male i byte o usare comandi incompatibili.

Unità 4.2 - Sniffing SPI e autenticazione

Tipo: laboratorio pratico.

Collegare il logic analyzer in parallelo a CS, CLK, MOSI, MISO e GND. Avviare la challenge dalla console DIVA e configurare CPOL/CPHA fino a ottenere byte stabili e coerenti.

Cosa osservare

- transizione di CS che delimita la transazione;
- opcode inviato su MOSI;
- indirizzo interno;
- dati restituiti su MISO;
- eventuale stringa usata nel confronto di autenticazione.

Un decoder che mostra dati casuali può indicare modalità clock errata, frequenza di acquisizione insufficiente o cablaggio MOSI/MISO invertito.

Unità 4.3 - Dump SPI e analisi firmware

Tipo: laboratorio pratico.

Il kit include una flash SPI separata e un'immagine `camera.bin` di riferimento. Il manuale usa il plugin:

```
1 ef> run spi.generic.readflash -w camera_dump.bin
```

L'immagine viene poi analizzata con uno strumento di riconoscimento dei formati e di estrazione:

```
1 binwalk -e camera_dump.bin
```

Consegna

Documentare dimensione, hash, firme individuate, filesystem o segmenti estratti e differenze rispetto a `camera.bin` quando si dispone di un dump reale.

Modulo 5 - BLE

Parte A - DIVA

Unità 5A.1 - Ricognizione BLE

Tipo: laboratorio guidato.

Attivare il laboratorio BLE dalla console DIVA, identificare l'interfaccia HCI e avviare la scansione:

```
1 ef> run ble.generic.scan -i 0
```

La consegna è identificare il dispositivo denominato EXPLIoT, registrando indirizzo, nome e qualità del segnale senza confondere dispositivi vicini.

Unità 5A.2 - Enumerazione GATT

Tipo: laboratorio guidato.

```
1 ef> run ble.generic.enum -a <BLEADDR> -s
2 ef> run ble.generic.enum -a <BLEADDR> -c
```

La prima operazione elenca i servizi; la seconda le caratteristiche. Per ogni elemento annotare UUID, handle, permessi e possibile funzione. Il manuale prosegue con scritture su un handle che controlla i LED: questa parte va interpretata come verifica funzionale del mapping GATT.

Unità 5A.3 - Dimostrazione di un handle vulnerabile

Tipo: dimostrazione controllata.

Il target DIVA contiene una challenge progettata per reagire in modo anomalo a valori troppo lunghi scritti sull'handle didattico. Non è un esercizio di scoperta indipendente: handle e comportamento sono già guidati dal manuale.

Il valore didattico consiste nell'osservare:

- comportamento corretto con payload previsto;
- soglia oltre la quale l'handle non risponde;
- necessità di validare lunghezza e formato lato periferica;
- differenza tra errore applicativo, disconnessione e crash.

Parte B - Smartwatch

Unità 5B.1 - Ricognizione

Tipo: laboratorio guidato.

Scansionare i dispositivi vicini, confrontare il risultato con l'indirizzo visualizzato dal watch e registrare l'identificatore corretto.

Unità 5B.2 - Enumerazione GATT

Tipo: laboratorio guidato.

Enumerare servizi e caratteristiche come per la DIVA. La tabella risultante serve da indice per le catture successive: UUID standard suggeriscono una funzione, mentre UUID custom richiedono correlazione sperimentale.

Unità 5B.3 - Acquisizione HCI su Android

Tipo: laboratorio di acquisizione.

Il manuale propone di abilitare il Bluetooth HCI snoop log nelle opzioni sviluppatore, riavviare Bluetooth e telefono, associare lo smartwatch all'app e compiere azioni come impostazione dell'ora, notifiche e misura del battito.

Il file `btsnoop_hci.log` viene poi trasferito al computer e aperto in Wireshark. La cattura è lato host Android: mostra i comandi HCI e il traffico GATT visto dallo stack del telefono, non una cattura radio passiva dell'intero ambiente.

Unità 5B.4 - Analisi GATT offline

Tipo: esercizio di analisi.

Il file fornito `Smartwatch.log` permette di svolgere l'attività senza telefono. Il filtro indicato dal manuale è:

```
1 btl2cap.cid == 0x0004
```

Obiettivo: individuare ATT Write Command, handle e payload e correlare ogni scrittura all'azione dell'app. L'esempio del manuale usa l'handle `0x0042` e un payload contenente campi di data e ora.

Unità 5B.5 - Controllo tramite GATT

Tipo: laboratorio pratico.

Dopo avere identificato handle e payload, il manuale usa il plugin generico di scrittura:

```
1 ef> run ble.generic.writechar -a <BDADDR> -n <handle> -w <valore>
```

Si verificano cambiamenti di ora/data e l'attivazione del sensore del battito. La parte di fuzzing è un'estensione controllata della challenge: modificare una sola porzione alla volta, registrare il risultato e interrompere in caso di riavvii ripetuti o modalità DFU.

Unità 5B.6 - LTK e decrittazione

Tipo: esercizio guidato su catture fornite.

Materiali:

- `ltk_exchange.pcap` contiene lo scambio utile al calcolo della LTK nella configurazione di esempio;
- `encrypted_known_ltk.pcap` contiene traffico cifrato;
- `ltk.txt` contiene il valore di riferimento;
- il README originale descrive la procedura prevista.

Il manuale usa `crackle` per produrre una cattura decifrata. L'obiettivo didattico è capire che la possibilità di recuperare la LTK dipende dal metodo di pairing e dalla disponibilità dei pacchetti

necessari; non tutte le sessioni BLE moderne sono decifrabili con la stessa procedura.

Modulo 6 - 802.15.4 e ZigBee

Unità 6.0A - Preparazione con una DIVA

Tipo: preparazione obbligatoria.

Con una sola DIVA si configura inizialmente il ruolo sender e, per il replay, si riconfigura la stessa scheda come receiver. PAN ID e indirizzi devono essere annotati perché verranno usati nei filtri e nei plugin successivi.

Unità 6.1 - Scoperta della rete

Tipo: laboratorio guidato.

Il manuale usa Zigbee Auditor e i plugin:

```
1 ef> run zbauditor.generic.deviceinfo
2 ef> run zbauditor.generic.nwscan
```

La consegna è identificare canale e PAN ID del proprio banco. Il canale è un prerequisito per lo sniffer.

Unità 6.2 - Cattura del traffico

Tipo: laboratorio di acquisizione.

```
1 ef> run zbauditor.generic.sniffer -c 12 -f cattura.pcap -t 10
```

Il valore 12 è soltanto l'esempio del manuale: usare il canale scoperto nell'unità precedente. In Wireshark verificare che la cattura contenga frame del proprio PAN e annotare indirizzi sorgente e destinazione.

Unità 6.3 - Replay controllato

Tipo: laboratorio pratico.

Filtrare i frame destinati alla propria DIVA, esportarli in un PCAP e ritrasmetterli con il plugin di replay indicato dal manuale:

```
1 ef> run zbauditor.generic.reply -c <canale> -f <cattura.pcap>
```

Il receiver deve essere in ascolto. Il risultato va valutato sulla console DIVA e nella cattura, distinguendo i frame ritrasmessi dagli acknowledgment.

Unità 6.0B - Preparazione con due DIVA

Tipo: preparazione alternativa.

Una DIVA resta sender e l'altra receiver. Entrambe usano lo stesso PAN ID; il sender deve essere configurato con l'indirizzo del receiver. Questa configurazione rende più chiara la separazione dei ruoli.

Unità 6.4 - Varianti con due DIVA

Tipo: variante, non nuovo esercizio.

Il manuale dichiara esplicitamente che scoperta, eavesdropping e replay sono gli stessi laboratori della configurazione con una DIVA. Le differenze operative sono:

- sender e receiver possono restare attivi contemporaneamente durante scoperta e cattura;
- prima del replay il sender originale va fermato, così il frame ricevuto può essere attribuito alla ritrasmissione.

Unità 6.5 - Decrittazione di catture ZigBee

Tipo: esercizio di analisi su dati forniti.

Materiali:

- `mi_pairing.pcap`: associazione del dispositivo e trasporto della chiave di rete;
- `mi_decrypt.pcap`: comunicazione applicativa successiva.

Il manuale descrive l'inserimento in Wireshark della chiave globale `ZigBeeAlliance09`, usata in alcuni sistemi per compatibilità. Questo consente di leggere il frame Transport Key della cattura di associazione e ricavare la Network Key. Inserendo anche la Network Key si può analizzare la cattura applicativa.

Obiettivo concettuale

Distinguere:

- Global Link Key nota;
- Network Key trasportata durante il join;
- cifratura dei frame applicativi;
- rischio derivante dall'uso di una chiave globale condivisa;
- differenza tra analisi offline della cattura fornita e replay su un banco autorizzato.

Come documentare ogni attività

Per ciascuna unità compilare:

- 1 Titolo:
- 2 Classificazione:
- 3 Hardware e software:
- 4 Schema dei collegamenti:

```
5 Parametri:
6 File di ingresso:
7 Procedura eseguita:
8 Risultato atteso:
9 Risultato osservato:
10 Evidenze salvate:
11 Errori e correzioni:
12 Conclusione di sicurezza:
```

Evidenze consigliate

- fotografia o schema del cablaggio;
- tabella dei pin;
- versione degli strumenti;
- comandi realmente eseguiti;
- hash SHA-256 di dump e catture;
- screenshot mirati;
- filtri Wireshark;
- confronto prima/dopo;
- interpretazione del rischio e contromisura.

Diagnosi rapida

Nessun dispositivo seriale

Controllare cavo dati, passthrough USB, `lsusb`, messaggi del kernel e presenza di sessioni seriali già aperte.

Testo UART illeggibile

Verificare baud rate, massa comune, incrocio TX/RX, livello logico e che la challenge UART sia attiva.

OpenOCD non rileva il target

Verificare alimentazione, pin, file di configurazione, versione di OpenOCD, reset e protezioni del microcontrollore.

Decoder I²C o SPI incoerente

Aumentare il sample rate del logic analyzer, verificare massa e canali, correggere CPOL/CPHA per SPI e assicurarsi che la challenge generi traffico.

BLE non compare

Verificare che il laboratorio BLE sia attivo, l'interfaccia HCI sia disponibile, il dispositivo non sia già connesso in modo esclusivo e la scansione usi l'interfaccia corretta.

ZigBee Auditor non cattura

Confermare il canale, il PAN ID, il ruolo della DIVA, la durata dello sniffer e la corretta enumerazione USB dell'Auditor.

Mappa dei materiali riorganizzati

I documenti originali sono in [documentazione](#). I file operativi sono stati spostati accanto alle unità che li usano:

- JTAGenum e relativo HEX nell'Unità 2.2;
- configurazioni OpenOCD nell'Unità 2.3;
- [camera.bin](#) nell'Unità 4.3;
- [Smartwatch.log](#) nell'Unità 5B.4;
- catture e LTK nell'Unità 5B.6;
- catture ZigBee nell'Unità 6.5.

I file originali non sono stati modificati; la riorganizzazione riguarda nomi delle cartelle, documentazione italiana e collocazione didattica.