Security Training Tutorial

Hands-on Security Training

- This tutorial describes the way in which IoTrain-Sim security training is conducted by using Cooja
- The key element of our approach is to run first a **reference simulation**, so that trainees understand the base simulation scenario
- This is followed by the deployment of malicious nodes in the reference network to create the **attack simulation**
  - Attacks are implemented by modifying source code files, thus resulting in an alteration of node behavior

Reference and Attack Scenarios

- IoTrain-Sim includes several training exercises
  - For each exercise, both the reference and attack scenarios are provided
  - Each exercise can be started via the corresponding menu entries in the IoTrain-Sim interface
- Our attack scenarios are based on information in
  - “RPL Attacks Framework” by A. D’Hondt et al. https://github.com/dhondta/rpl-attacks/blob/master/doc/report.pdf
- Since the attack mechanisms exploit routing protocol weaknesses, we recommend consulting also the “Routing Protocol Overview” section

Attack Scenario Overview

Name of attack	Harmful effects	Coping strategies	Strategy issues
Flooding attack	Increased load and energy consumption	Global or local fix	Increase of repair time and network delay
DODAG version attack	Increased delay; reset of network topology	Lightweight IDS	None
Blackhole attack	High packet loss rate; control traffic overhead; increased delay	Trusted perception mechanism; detection techniques	Difficult to resist multiple attacks
Decreased rank attack	Low packet rate; control traffic overhead; increased delay	Version hash; validation; use of random numbers	Cannot handle dynamic attacks

Training Flow

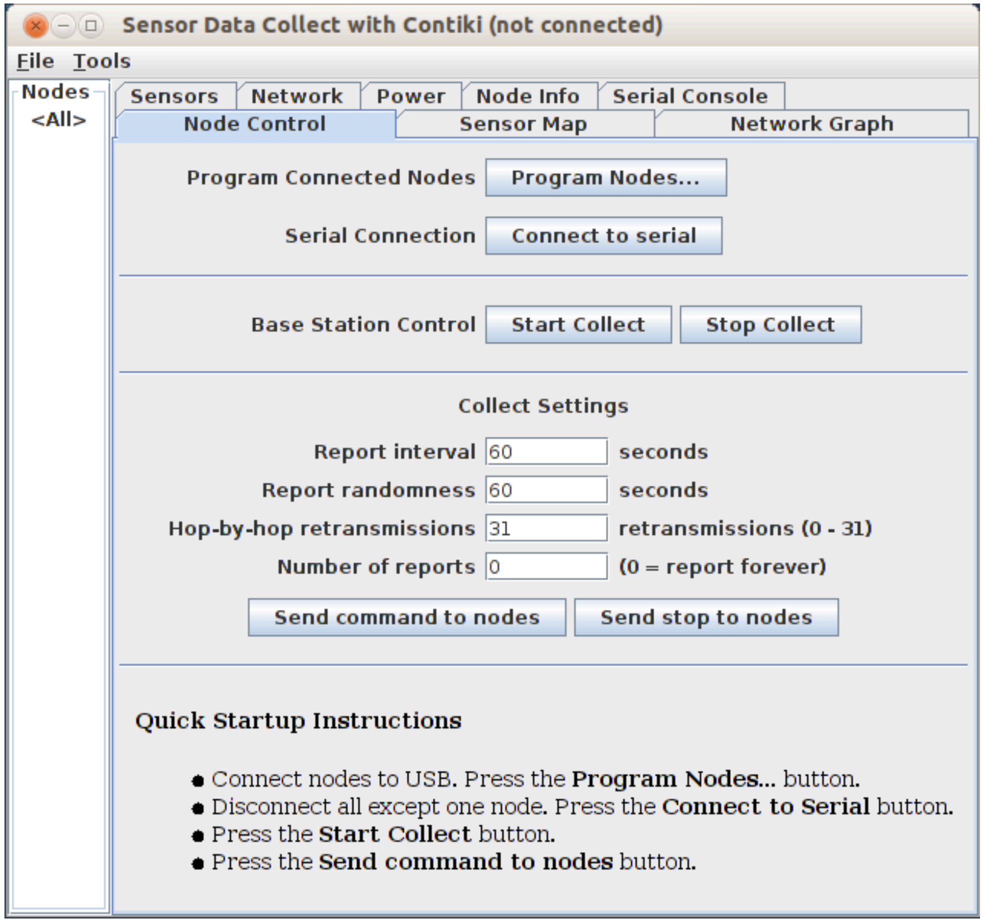
1. Start the **reference simulation** for the desired security attack
2. Use the “Collect View” tool on the sink node to collect data and understand the reference behavior
3. Start the **attack simulation** for the same security attack
4. Use again the “Collect View” tool on the sink node to collect data and understand the attack behavior

What is Collect View

- Collect View is a Java based application in Contiki used for internal mote information visualization
- A mote is acting as a sink, receiving traffic, while the other motes are acting as traffic sources
    - Source motes send important parameters to the sink
- Collect View uses a Graphical User Interface (GUI) for visualizing mote parameters
    - In an attack simulation, this tool will be used to observe the impact of malicious nodes on the network

Running Collect View

- To open Collect View, run the following commands
  - cd contiki/tools/collect-view
  - ant run
- The interface with the Node Control panel selected will be displayed, as illustrated in the screenshot on the right



Collect Settings

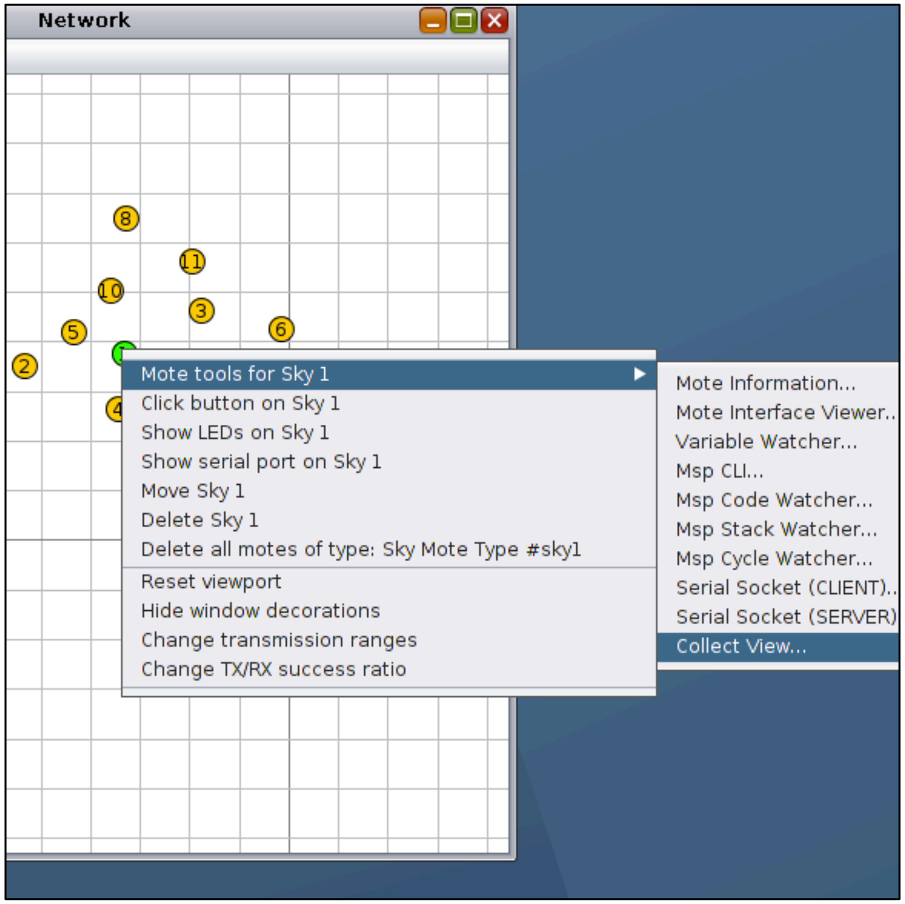
Report interval 60 seconds
Report randomness 60 seconds
Hop-by-hop retransmissions 31 retransmissions (0 - 31)
Number of reports 0 (0 = report forever)

Quick Startup Instructions

- Connect nodes to USB. Press the **Program Nodes...** button.
- Disconnect all except one node. Press the **Connect to Serial** button.
- Press the **Start Collect** button.
- Press the **Send command to nodes** button.

Using Collect View

- To use Collect View, do the following
  1. Find the sink node
  2. Right click on the sink node, then select Mote tools for … > Collect View
  3. In the Node Control panel, click on "Start Collect", then click on "Send command to nodes"



Creating New Scenarios

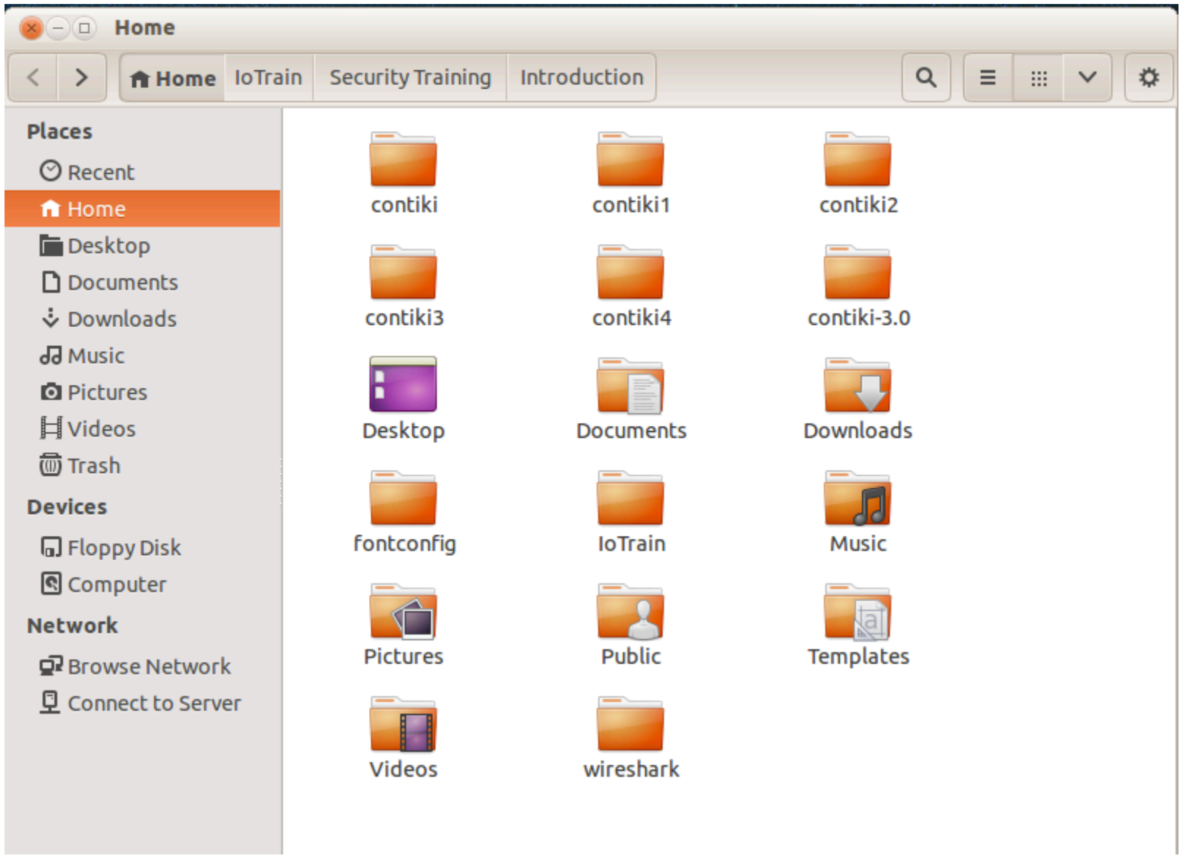
Reference Simulation

- Open Cooja and click on File > New simulation
- Create the mote types that will make up the network
  - Typically, the reference network has two types of motes
    - One sink mote, which would function as an LBR and DODAG router
    - Several leaf motes, functioning as mere sensor data collectors
  - Motes are usually based on the following firmware files
    - Sink more → “contiki/examples/ipv6/rpl-collect/sink.c”
    - Leaf motes → “contiki/examples/ipv6/rpl-collect/udp-sender.c”
- Save the simulation as a CSC file

Attack Simulation

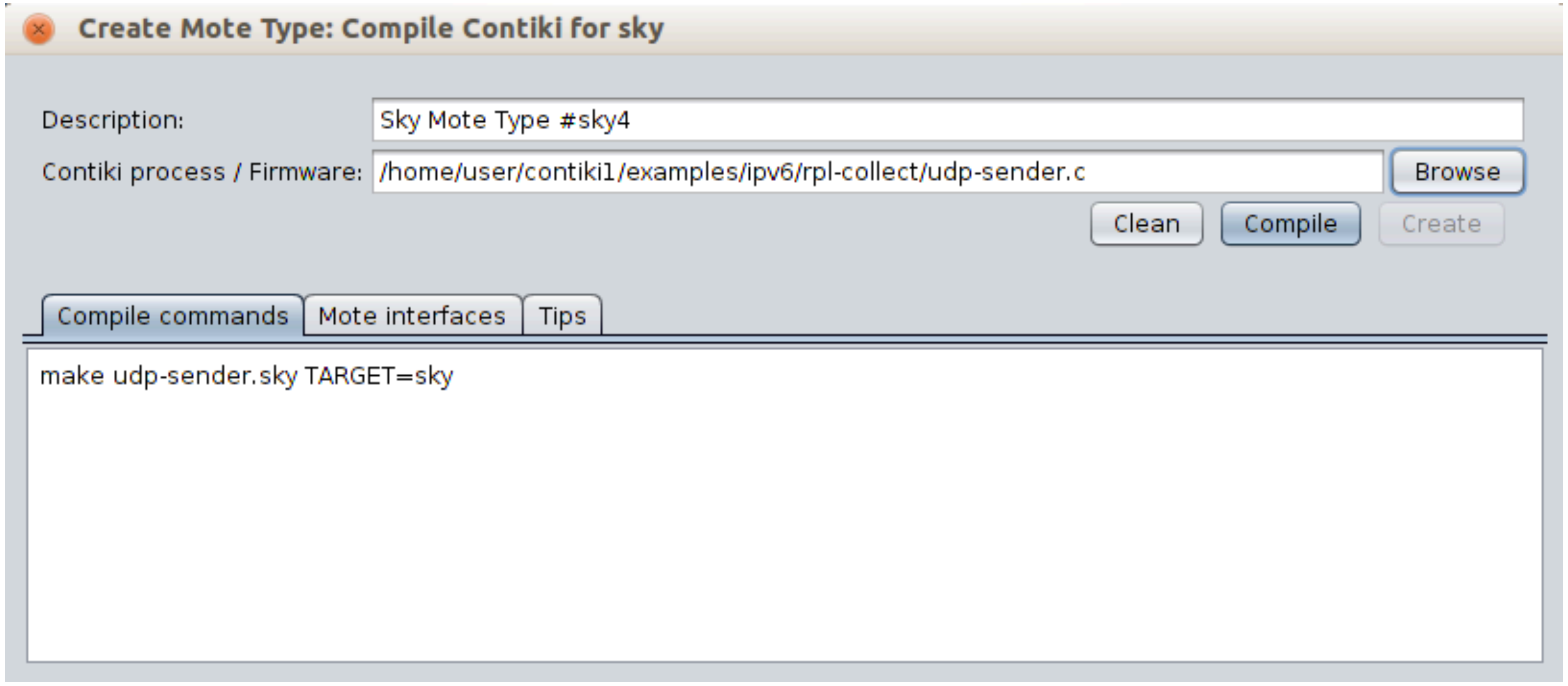
- Attack simulation is implemented by modifying the behavior of one or more motes, without altering the normal behavior of the other nodes
  - Thus, one can assess network changes during security attacks
- The recommended method to achieve this is
  1. Duplicate the “contiki/” folder to create a new Contiki instance (for example, you can use “contiki1/” for flooding attack, “contiki2/” for version number attack, etc.)
  2. Modify the necessary files in the new Contiki instance according to the specificities of the attack
  3. Open the target reference simulation file in Cooja
  4. Assign the node firmware within the new Contiki instance to one of the motes in the reference scenario, to give it a malicious behavior

Multiple Contiki Folders



Several Contiki folders used to create different types of malicious motes

Malicious Mote Configuration



Description: Sky Mote Type #sky4

Contiki process / Firmware: /home/user/contiki1/examples/ipv6/rpl-collect/udp-sender.c

```
make udp-sender.sky TARGET=sky
```

Creating a malicious mote based on source code from another Contiki instance