CEng 491 -- PENIOT KickOff Document

"*PENIOT: A Penetration Testing Tool for Internet of Things*" KickOff Document

Description

PENIOT is an extensible penetration testing tool for the Internet of Things (IoT). It is a penetration testing tool that automates the process of security testing for various IoT devices and network configurations by launching selected IoT-specific attacks on devices through an easy-to-use menu with attack parameters set by the user. The end product will be a penetration testing tool that IoT software and hardware developers can utilize to test the security of their products against common/new attacks on IoT-specific protocols and devices. The tool can also be extended by security engineers to integrate additional tests as new IoT devices/protocols and corresponding exploits are launched.

Master feature list

MF-1 - Sniffing Zigbee message traffic
MF-2 - Zigbee attacks (at least 1)
MF-3 - Sniffing MQTT message traffic[1]
MF-4 - MQTT attacks (at least 1)
MF-5 - Sniffing RPL message traffic
MF-6 - RPL attacks[2] (at least 1)
MF-7 - Sniffing BLE message traffic[3]
MF-8 - BLE attacks (at least 1)
MF-9 - Easy-to-use menu (interface)
MF-10 - Report generation with respect to test cases
MF-11 - External module integration capability

[1] MQTT: Message Queuing Telemetry Transport
[2] RPL: Routing over Low Power and Lossy Networks
[3] BLE: Bluetooth Low Energy

Workpackages

WP #	Term	WP title (this should be as short and as descriptive as possible)	Estimated number of person-months
1	491	Project planning and architecture design	3
2	491	Project tool planning, obtaining and deployment	2
3	491	Zigbee attacks	7
4	492	BLE (Bluetooth Low Energy) attacks	6
5	492	MQTT (MQ Telemetry Transport) attacks	4
6	492	RPL (Routing over Low Power and Lossy Networks) attacks	4
7	492	GUI Implementation, report generation and external module integration structure	4
8	492	Implementation of additional attacks and countermeasures (Bonus)	3
		Total:	30+3

Detailed Descriptions of High-Level Workpackages

WP1 - Project planning and architecture design

1. Develop the list of master features of the project.
2. Produce project development plan in accordance with Master Feature List.
3. Design the overall architecture of the project.
4. Analyze risks and make a management plan.

WP2 - Project tool planning, obtaining and deployment

1. Decide the software and hardware tools that will be required to implement this project.
2. Plan how we will get these tools.
3. Obtain and deploy the tools that can be obtained, order the ones that needs to be ordered.
4. Decide what to do if some of the tools cannot be obtained.

WP3 - Zigbee attacks

1. Analyze Zigbee protocol and find possible vulnerabilities so that we could implement the attacks below (MF-1, MF-2)
    a. MF-1 (Sniffing): Includes packet sniffing (valid data) and sending them repeatedly or with delay for malicious reasons

    b. MF-2 (Zigbee attacks): Replay attack, denial of service attack, man in the middle attack,etc.
2. Design the overall module architecture for Zigbee
3. Implement Zigbee attacks
4. Test the implemented attacks

WP4 - BLE (Bluetooth Low Energy) *attacks*

1. Analyze BLE protocol and find possible vulnerabilities so that we could implement above attacks (MF-7, MF-8)
    a. MF-7 (Sniffing): Includes sniffing a valid BLE communication and trying to get useful vulnerability information.
    b. MF-8 (BLE attacks): Attacks on the pairing (connection setting) algorithm, replay attack, fuzzing attack,etc.
2. Design the overall module architecture for BLE
3. Implement BLE attacks
4. Test the implemented attacks

WP5 - MQTT (Message Queuing Telemetry Transport) attacks

1. Analyze MQTT protocol and find possible vulnerabilities (MF-3,MF-4)
    a. MF-3 (Sniffing): Includes sniffing a valid MQTT communication and trying to get useful vulnerability information.
    b. MF-4 (MQTT attacks): Replay attack, fuzzing attacks, denial of service (DOS) attack,etc.
2. Design the overall module architecture for MQTT
3. Implement MQTT attacks
4.Test the implemented attacks

WP6 - RPL attacks

1. Analyze RPL protocol and find possible vulnerabilities so that we could implement above attacks
    a. MF-5 (Sniffing): Includes eavesdropping on a RPL communication in order for analyzing message content or learning network structure.
    b. MF-6 (RPL attacks): Flooding Attacks, Version Number Attacks, Wormhole Attacks, Routing Information Replay Attacks, DAO Inconsistency Attacks in Storing Mode and the like.
2. Design the overall module architecture for RPL
3. Implement RPL attacks
4. Test the implemented attacks

WP7 - GUI Implementation, report generation and external module integration

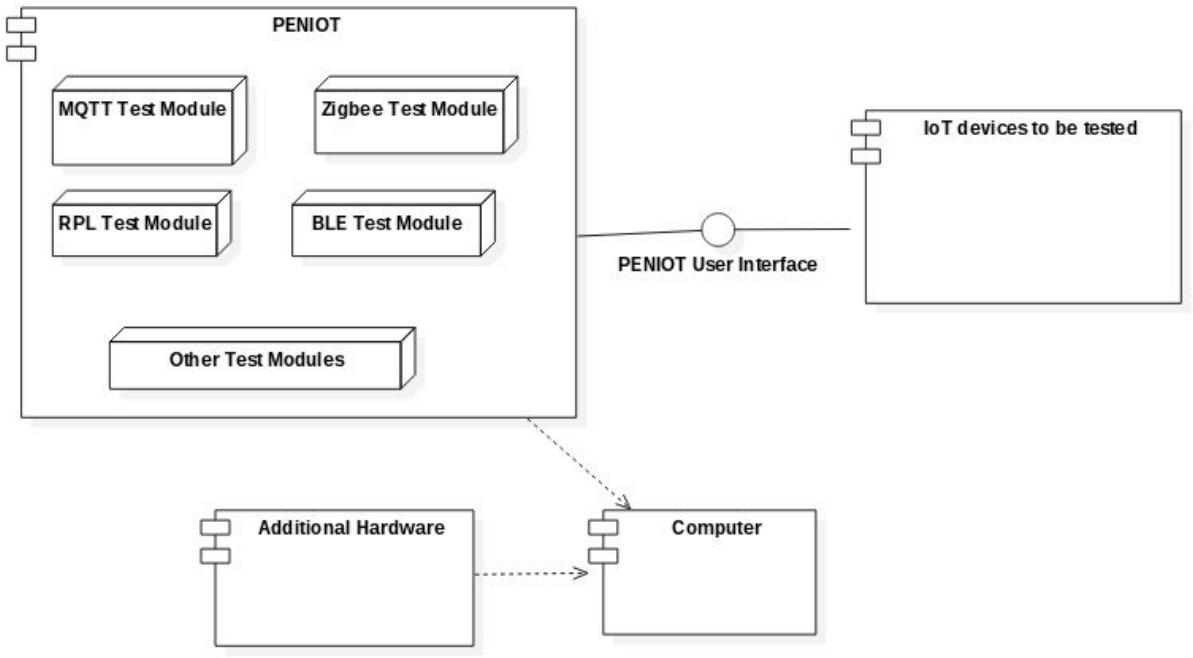
1. Design automated test interface
2. Design user defined attack interface
3. Connect implementations to interfaces
4. Design external module integration capabilities
5. Design and implement the report generation

6. Test the user interface

WP8 - Implementation of additional attacks and counter measures (Bonus)

1. Although it is not an aim of our project, if we find a novel (or better than any existing ones) countermeasure for any of the vulnerabilities, then we may implement (and test) that and prepare a paper for that.
2. If we finish our project early, we could test other protocols as well such as 6LowPan.

Overall Systems Architecture



In our design, there will be one module for each different protocol such as MQTT or Zigbee. This enables us to extend PENIOT with additional attacks for other protocols as well. Since these protocols use wireless connection, we may need to have additional devices to send or receive radio signals. Moreover, we provide an easy-use interface to use Peniot efficiently.

TimeLine

Risk Assessment

Risk #	Description	Possible Solution(s)
1	Obtaining necessary hardware devices to develop and test our tool may not be always possible, especially for less commonly used protocols.	We may use Amazon IoT Device Simulator to develop and test such tools.
2	Since this project is a very novel and experimental project (there are very few IoT testing tools, and the existing ones are at a very preliminary stage) and there is a great diversity of IoT technologies, we may fail to implement some tests.	We may replace some of these failed tests with some other tests on different IoT devices and protocols.
3	The implementation of some attacks may take longer than planned due to attack complexity/delays in required equipment setup	The schedule can be adjusted to split work among group members to do development of multiple attacks in parallel