P-Scan – Test Case Specification

Version	Date	DESCRIPTION	Author
1.0	08/11/2019	Creation	Jonathan Pauc/ Jérôme Hamel

Table des matières

Introduction

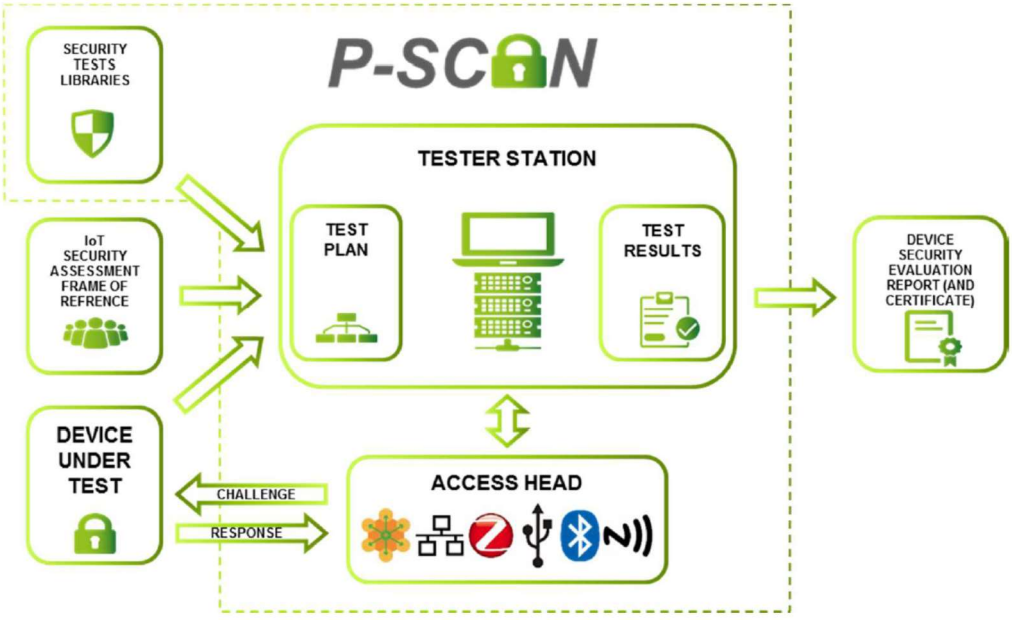
Scope

The purpose of this document is to define the perimeter covered by the security tests of P-SCAN.

Each Test is described as per below.

FIELD	DESCRIPTION
Name	Denomination of the Test Case
Description	Description of the purpose and the objective of the Test Case
Test scenario	Description of the Test Case flow and its different steps
Expected behavior	Description of the expected behavior from the device at each steps
Success oracle	Criteria to decide the test verdict
Related weaknesses	Brief explanation of the weaknesses related to the tested vulnerabilities
References	Technical references and vulnerabilities covered by the Test Case
DUT/SUT prerequisites	Conditions to reach before executing the Test Case
Solutions and mitigations	What actions or approaches are recommended to mitigate this failing test

P-SCAN introduction



Bureau Veritas P-SCAN is a Vulnerability Scanner for the following communication Channels.

- Wifi
- Bluetooth Low Energy
- Zigbee

P-SCAN check IoT devices in a BlackBox approach against the key known vulnerabilities used by hackers.

No preparation is needed from the device vendor.

P-SCAN service provide an immediate feedback on the communication channel vulnerabilities that are present on the device that can be used by attackers.

P-SCAN can also be used as part of wider cyber security assessment

P-SCAN tester access the DUT over the air interface via access heads implementing the protocol layers from each communication channel

References

To be added

Definition, acronyms and abbreviations

IOT device:
IOT device connects to a network. An IOT device may contain software, hardware,
Sensors

Black box testing:
Testing method that examines the functionality of a DUT without specific knowledge of the DUT's code/internal structure. The tester is aware of *what* (either exposed or hidden functionalities) the DUT is supposed to do but is not aware of *how* it does.

Common vulnerabilities and exposures:
(CVE®)

Communication channel:
Specific association of a given hardware interface and a given communication protocol layer.

Communication protocol:
System of rules, expressed by algorithms and data structures that allow information exchange between devices. Protocols are to communications what algorithms are to computations. Communication Protocols are built on a layered software model.

DUT:
Device Under Test.

Evaluation report:
Report generated at the end of the evaluation phase. The evaluation report lists all the test cases executed towards the DUT and the associated verdict.

Hardware interface:
Physical media used to connect DUT. May be wired or wireless (for RF communications). For example: usb, ethernet, 802.15.4, hardware interfaces are hosted by an access head platform

Ble:
Bluetooth low energy

BLUETOOTH LOW ENERGY

Test Cases	
BLE#1	FCH_CKM_: Connection mode
BLE#2	FCH_CKM_: Diffie Hellman key exchange
BLE#3	FCH_COP: Legacy OOB connection.
BLE#4	FCH_COP_ : SMP Pairing timeout
BLE#5	FCH_SECURE_CHANNEL_: SMP downgrade pairing
BLE#6	FCH_DATA: Gatt check write on read only characteristics values.
BLE#7	FCH_DATA: Gatt check notifications
BLE#8	FCH_DATA: Gatt Check read permissions with read characteristic by services
BLE#9	FCH_DATA_: Gatt check read permissions with read characteristic.
BLE#10	FCH_DATA_: Check Read permissions with read long characteristic by services.
BLE#11	FCH_DATA: Gatt check read permissions with read long characteristic
BLE#12	FCH_DATA: Check read permissions with read multiple by services.
BLE#13	FCH_DATA_: Check read permissions with read using UUID by services.
BLE#14	FCH_CONFIGURATION_POLICY: Gatt check write on property attributes
BLE#15	FCH_CONFIGURATION_POLICY: Gatt check write on writable characteristic values
BLE#16	FCH_CONFIGURATION_POLICY: Check write on characteristics descriptors
BLE#17	FCH_CONFIGURATION_POLICY_: Check list of services
BLE#18	CH_CONFIGURATION_POLICY: Advertissing data
BLE#19	FCH_RUNTIME_SECURE: Dos connecting
BLE#20	FCH_IDENTIFICATION_SECURE: Advertissing with public address

BLE#1 - FCH_CKM_: Connection mode

FIELD	DESCRIPTION
Name	SMP Connection mode
Description	This test case checks if Secure Connection pairing is supported. It takes as inputs the DUT address and its type
Test scenario	This test sends a pairing request (with Secure Connection flag enabled), then it checks that the DUT accept the pairing with the Secure Connection flag enabled. **Note** :that this test does not check the secure connection implementation. It only checks that the DUT pretends to support this mode.
Expected behavior	Expected behavior: The device should support Secure Connection pairing. In this test case, legacy pairing - especially without OOB - is considered unsecured.
Success oracle	Success if returns True (secure connection mode is supported)
Related weaknesses	Key derivations should not happen from weak user pin/password
References	Bluetooth Core Specs V5.0 CVE 2018-9119
DUT/SUT prerequisites	Device ready to pair
Solutions and mitigations	Implement a secure connection

BLE#2 - FCH_CKM_: Diffie Hellman key exchange

FIELD	DESCRIPTION
Name	Diffie Hellman key exchanged
Description	This test case checks if DUT is vulnerable to the CVE 2018-5383 (Fixed Coordinate Invalid Curve Attack).This CVE affects devices which does not sufficiently validate elliptic curve parameters used to generate public keys during a Diffie-Hellman key exchange. This may allow a remote attacker to obtain the encryption key used by the device.
Test scenario	This test sends a pairing request with Y-coordinate of the Public Key equal to 0 then it checks if the DUT accept the pairing. Note: Test prerequisite is to have Secure Connection support (cf. FCh_CKM_BLE#1)
Expected behavior	The device should refuse the pairing with PKy=0
Success oracle	Success if returns True
Related weaknesses	Wrong implementation of Diffie Hellman key exchanged
References	Bluetooth Core Specs V5.0 CVE 2018-5383
DUT/SUT prerequisites	Device ready to pair
Solutions and mitigations	Implement a secure connection

BLE#3 FCH_COP: Legacy OOB connection.

FIELD	***DESCRIPTION***
Name	Legacy OOB connection
Description	When Secure pairing is not supported, legacy pairing can be considered but it must support Out Of Band (OOB) pairing. In other cases (without OOB) Legacy pairing mode is considered unsecured.
Test scenario	This test case send a pairing request in legacy mode with OOB, then it checks the DUT response for OOB support. **Note**: A true verdict does not mean anything on the implementation of the legacy connection mode with OOB support. It only means that the DUT pretends to support the oob pairing.
Expected behavior	*In legacy pairing mode, the device must support pairing with OOB mechanisms.*
Success oracle	Success if returns True (secure connection mode is supported)
Related weaknesses	Key derivations should not happen from weak user pin/password
References	Bluetooth Core Specs V5.0 CVE 2018-9119
DUT/SUT prerequisites	Device ready to pair
Solutions and mitigations	Implement a secure connection

BLE#4 FCH_COP_ : SMP Pairing timeout

FIELD	DESCRIPTION
Name	SMP Pairing timeout
Description	The Bluetooth specification (v5.0) mentions a 30 seconds timeout during pairing procedure (Vol 3. Part H 3.4). After 30s the pairing should failed in order to protect from some attacks.
Test scenario	This test case checks that the DUT implement such timeout mechanism. This test case initiate a legacy pairing, then a sleep of 31 seconds is introduced between the pairing request and confirm request. Finally the test checks that the pairing procedure is canceled.
Expected behavior	The device must stops the pairing procedure after 30sec and reject the confirm request
Success oracle	Success if returns True (pairing fails)
Related weaknesses	If the window is too large it makes easier some attacks
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to pair
Solutions and mitigations	Implement a secure connection

BLE#5 FCH_SECURE_CHANNEL_: SMP downgrade pairing

FIELD	DESCRIPTION
Name	SMP Downgrade Pairing
Description	The Bluetooth specification mentions that a peripheral has to accept the legacy pairing mode if the client does not support the secure connection mode. However the legacy mode (without OOB) is considered not secure, that's why this test fails if the DUT accepts to downgrade the pairing mode (without OOB).
Test scenario	This test case first sends a pairing request with Secure Connection flag enabled, then it sends fake random and confirm. Finally it sends a new pairing request without Secure Connection flag and then checks that the pairing response keeps the Secure Connection flag enabled (if not the test fails).
Expected behavior	The device must rejects the downgraded mode for security reasons. Bluetooth specifications mentions that a peripheral has to accept the legacy pairing mode if the client does not support the secure connection mode. However, the legacy mode is considered not secure
Success oracle	Success if returns True (secure connection mode downgrade rejected)
Related weaknesses	Legacy mode (without OOB) considered as non-secure
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to pair
Solutions and mitigations	Implement a secure connection

BLE#6 - FCH_DATA: Gatt check write on read only characteristics values.

FIELD	DESCRIPTION
Name	Gatt check write on read only characteristics values
Description	This test performs write(s) on non writable properties on primary services without any pairing.
Test scenario	This test case starts with the scan of GATT services. Then it retrieves all properties not writable without authentication Finally for each of these read-only properties, it tries to perform a write.
Expected behavior	The device must reject any write command on read only property
Success oracle	Success if returns True
Related weaknesses	Data integrity
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to connect
Solutions and mitigations	Implement a secure connection

BLE#7 - FCH_DATA: Gatt check notifications

FIELD	DESCRIPTION
Name	*Gatt check notifications*
Description	The test connects to the device without pairing and waits for notifications during 10 seconds
Test scenario	*The test connects to the DUT on ATT layer, without pairing, and waits for notifications during 10 seconds.* *The test case checks that during this time lapse, the DUT does NOT notify (and possibly leak) information.*
Expected behavior	*No notifications are publicly sent by the device*
Success oracle	*Success if returns True*
Related weaknesses	*Information exposure*
References	*Bluetooth Core Specs V5.0*
DUT/SUT prerequisites	*Device ready to connect*
Solutions and mitigations	*Implement a secure connection before notifications*

BLE#8 - FCH_DATA: Gatt Check read permissions with read characteristic by services

FIELD	DESCRIPTION
Name	Gatt check Read permissions with read characteristic by services
Description	Non authenticated read, can lead to potential leak of information. This test checks if it is possible to read BLE characteristics without authentication.
Test scenario	This test case starts with the scan of GATT services. Then it checks if it's possible to read a characteristics value using the "Read Characteristic Value" method (Bluetooth specs V5.0, Vol3, Part G, chapter 4.8.1). The test fails if it is possible to read information without authentication. **Note**: A failed verdict may trigger a potential leak of information through a simple read. However it does not necessarily mean a vulnerability for the DUT. If value(s) read are public and considered not confidential, there may be no issue.
Expected behavior	The DUT does NOT allow read command without preliminary pairing
Success oracle	Success if returns True
Related weaknesses	Information exposure
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to connect List of services expected available
Solutions and mitigations	Implement a secure connection

BLE#9 - FCH_DATA_: Gatt check read permissions with read characteristic.

FIELD	DESCRIPTION
Name	Gatt check Read permissions with read characteristic
Description	Non authenticated read, can lead to potential leak of information. This test checks if it is possible to read BLE characteristics without authentication.
Test scenario	This test case takes as input a range of handles. For each of these handles, it checks if it's possible to read a value using the "Read Characteristic Value" method (Bluetooth specs V5.0, Vol3, Part G, chapter 4.8.1). The test fails if it is possible to read information without authentication. Note: A failed verdict may trigger a potential leak of information through a simple read. However it does not necessarily mean a vulnerability for the DUT. If value(s) read are public and considered not confidential, there may be no issue.
Expected behavior	The device does not allow read command without pairing
Success oracle	Success if returns True
Related weaknesses	Information exposure
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to pair List of services expected available
Solutions and mitigations	Implement a secure connection

BLE#10 - FCH_DATA_: Check Read permissions with read long characteristic by services.

FIELD	DESCRIPTION
Name	Gatt check Read permissions with read long characteristic by services
Description	Non authenticated read, can lead to potential leak of information. This test checks if it is possible to read BLE characteristics without authentication.
Test scenario	This test case starts with the scan of GATT services. Then it checks if it's possible to read a characteristics value using the "Read Long Characteristic Values" method (Bluetooth specs V5.0, Vol3, Part G, chapter 4.8.3). The test fails if it is possible to read information without authentication. Note: A failed verdict may trigger a potential leak of information through a simple read. However it does not necessarily mean a vulnerability for the DUT. If value(s) read are public and considered not confidential,there may be no issue.
Expected behavior	The DUT does NOT allow read command without preliminary pairing
Success oracle	Success if returns True
Related weaknesses	Information exposure
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to connect List of services expected available
Solutions and mitigations	Implement a secure connection

BLE#11 - FCH_DATA: Gatt check read permissions with read long characteristic

FIELD	DESCRIPTION
Name	Gatt check Read permissions with read long characteristic
Description	Non authenticated read, can lead to potential leak of information. This test checks if it is possible to read BLE characteristics without authentication.
Test scenario	This test case takes as input a range of handles. For each of these handles, it checks if it's possible to read a value using the "Read Long Characteristic Values" method (Bluetooth specs V5.0, Vol3, Part G, chapter 4.8.3). The test fails if it is possible to read information without authentication. Note: A failed verdict may trigger a potential leak of information through a simple read. However it does not necessarily mean a vulnerability for the DUT. If value(s) read are public and considered not confidential, there may be no issue.
Expected behavior	The device does not allow read command without pairing
Success oracle	Success if returns True
Related weaknesses	Information exposure
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to connect List of services expected available
Solutions and mitigations	Implement a secure connection

BLE#12 - FCH_DATA: Check read permissions with read multiple by services.

FIELD	DESCRIPTION
Name	Gatt Check Read permissions with read multiple by services
Description	Non authenticated read, can lead to potential leak of information. This test checks if it is possible to read BLE characteristics without authentication.
Test scenario	This test case starts with the scan of GATT services. Then it checks if it's possible to read a characteristics value using the "Read Multiple Characteristic Values" method (Bluetooth specs V5.0, Vol3, Part G, chapter 4.8.4). The test fails if it is possible to read information without authentication. Note: A failed verdict may trigger a potential leak of information through a simple read. However it does not necessarily mean a vulnerability for the DUT. If value(s) read are public and considered not confidential, there may be no issue.
Expected behavior	The DUT does NOT allow read command without preliminary pairing
Success oracle	Success if returns True
Related weaknesses	Information exposure
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to connect
Solutions and mitigations	Implement a secure connection

BLE#13 - FCH_DATA_: Check read permissions with read using UUID by services.

FIELD	DESCRIPTION
Name	Gatt Check Read permissions with read using UUID by services
Description	Non authenticated read, can lead to potential leak of information. This test checks if it is possible to read BLE characteristics without authentication
Test scenario	This test case starts with the scan of GATT services. Then it checks if it's possible to read a characteristics value using the "Read Using Characteristic UUID" method (Bluetooth specs V5.0, Vol3, Part G, chapter 4.8.2). The test fails if it is possible to read information without authentication. ***Note**: A failed verdict may trigger a potential leak of information through a simple read. However it does not* necessarily mean a vulnerability for the DUT. If value(s) read are public and considered not confidential, there may be no issue.
Expected behavior	The DUT does NOT allow read command without preliminary pairing
Success oracle	Success if returns True
Related weaknesses	Leak of information
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to connect
Solutions and mitigations	Implement a secure connection

BLE#14 - FCH_CONFIGURATION_POLICY: Gatt check write on property attributes

FIELD	DESCRIPTION
Name	Gatt check write on property attributes
Description	The presence of writable properties (without authentication) could allows to spy a write and eventually do a replay.
Test scenario	This test case starts with the scan of GATT services. Then it checks that the DUT does NOT expose property attributes in writable mode. No write is performed: the test only checks the properties exposed. ***Note**: A false verdict may trigger a potential security issue. However a write does not necessarily mean a vulnerability for the DUT. An exchange with the developer could be realized to verify if the write mode is necessary.*
Expected behavior	The device must always set the write with authentication
Success oracle	Success if returns True
Related weaknesses	Data integrity
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to connect
Solutions and mitigations	Implement a secure connection

BLE#15 - FCH_CONFIGURATION_POLICY: Gatt check write on writable characteristic values

FIELD	DESCRIPTION
Name	Gatt check write on writable characteristic values
Description	This test tries to write without any authentication. If possible such mechanism could lead to security problems.
Test scenario	This test case starts with the scan of GATT services then it gets readable and writable properties (to minimize the risk of breaking the DUT). Finally it checks that the write of the property is not possible. **Note**: A false verdict may trigger a potential security issue. However a write does not necessarily mean a vulnerability for the DUT. An exchange with the developer could be realized to verify if the write mode is necessary.
Expected behavior	The device must rejects the write commands with an error.
Success oracle	Success if returns True
Related weaknesses	Data integrity
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to connect
Solutions and mitigations	Implement a secure connection

BLE#16 - FCH_CONFIGURATION_POLICY: Check write on characteristics descriptors

FIELD	DESCRIPTION
Name	Gatt Check write on characteristics descriptors
Description	This test tries to write without any authentication. If possible such mechanism could lead to security problems.
Test scenario	This test case starts with the scan of GATT services then it tries to overwrite all characteristics descriptors found. The test checks that the write is not possible. ***Note1**: To avoid breaking the DUT, a read is first performed to restore the device in case of successful write* ***Note2**: A false verdict may trigger a potential security issue. However a write does not necessarily mean a vulnerability for the DUT. An exchange with the developer could be realized to verify if the write mode is necessary.*
Expected behavior	The device must rejects the write commands with an error.
Success oracle	Success if returns True
Related weaknesses	Data integrity
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to connect
Solutions and mitigations	Implement a secure connection

BLE#17 - FCH_CONFIGURATION_POLICY_: Check list of services

FIELD	DESCRIPTION
Name	Gatt Check List of services
Description	This test aims to ensure that the manufacturer is aware of the exposed services.
Test scenario	This test case starts with the scan of GATT services. Then the test case compares it with the expected list of services given as input.
Expected behavior	The DUT only exposes the list of services expected
Success oracle	Success if returns True
Related weaknesses	Leak of information
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to pair List of services expected available
Solutions and mitigations	Implement a secure connection

BLE#18 - CH_CONFIGURATION_POLICY: Advertissing data

FIELD	DESCRIPTION
Name	Advertising data
Description	The test checks if the device broadcasts proprietary data
Test scenario	The test case scans available devices. It then checks if proprietary data is being advertised by the DUT (address taken as input). If the DUT is not available or not detected, the test returns an INCONCLUSIVE verdict. Note: A false verdict may trigger a potential security hole. However, it does not necessarily mean a vulnerability for the DUT. If data advertised are public and considered not confidential there may be no issue.
Expected behavior	The device does not broadcast proprietary data
Success oracle	Success if returns True
Related weaknesses	Leak of information
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to advertise
Solutions and mitigations	Implement a secure connection

BLE#19 - FCH_RUNTIME_SECURE: Dos connecting

FIELD	DESCRIPTION
Name	Dos connecting
Description	This test checks if it is possible to perform a denial of services though repetitive connections.
Test scenario	In a first step, during a default polling time (or a time given as input) the test counts the number of advertising packets received from the DUT. In a second step, the test tries to keep connected to the DUT as much as possible. Simultaneously, it scans and counts the number of advertising packets received during the same first step time. The test returns a true verdict if the ratio between the number of advertising packets received during the second step and the first step is under a default limit equal to 50% (or a limit given as input). The test returns false if more packets than 100 - default ratio are considered dropped.
Expected behavior	The device integrates a method to avoid this kind of DOS
Success oracle	Success if returns True
Related weaknesses	Denial of service
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to pair List of services expected available
Solutions and mitigations	Implement a secure connection

BLE#20 - FCH_IDENTIFICATION_SECURE: Advertissing with public address

FIELD	DESCRIPTION
Name	Advertising with public address
Description	The test checks if the device broadcasts with public address
Test scenario	A scan is launched and if the DUT is not available or not detected, the test returns an INCONCLUSIVE verdict. The test returns a SUCCESS verdict if address is not public Note: A false verdict indicates that the DUT address is public. It does not mean a vulnerabilty for the DUT,but public addresses make easier a range of attacks and make possible geo tracking. Hence it is not considered safe for some applications.
Expected behavior	The device broadcasts does NOT broadcast with public address
Success oracle	Success if returns True
Related weaknesses	Spoofing
References	Bluetooth Core Specs V5.0
DUT/SUT prerequisites	Device ready to advertise
Solutions and mitigations	Implement a secure connection

WIFI

Test Cases	
WIFI#1	FCH_CKM: TKIP-PTK Reinstallation Attack / Delayed Plaintext Message 3
WIFI#2	FCH_CKM: TKIP-PTK Reinstallation Attack / Consecutive Plaintext Message 3
WIFI#3	FCH_CKM: TKIP-PTK Reinstallation Attack / Consecutive Encrypted Message 3
WIFI#4	FCH_CKM: TKIP-PTK Reinstallation Attack / Plaintext and Encrypted Message 3
WIFI#5	FCH_CKM: TKIP-GTK Reinstallation Attack in Group Key Handshake
WIFI#6	FCH_CKM: TKIP-GTK Reinstallation Attack in the 4-way Handshake
WIFI#7	FCH_CKM: TKIP-IGTK Reinstallation Attack in Group Key Handshake
WIFI#8	FCH_CKM: TKIP-IGTK Reinstallation Attack in the 4-way Handshake
WIFI#9	FCH_COP: Check for Obsolete Protocols
WIFI#10	FCH_COP: Group Message Replay
WIFI#11	FCH_ IDENTIFICATION_SECURE _: Broadcast Deauthentication Attack

Wifi Tests should be executed in a Faraday shield or in an environment with no other Wifi access points.

WIFI#1 - FCH_CKM: TKIP-PTK Reinstallation Attack / Delayed Plaintext Message 3

FIELD	DESCRIPTION
Name	*TKIP-PTK Reinstallation Attack / Delayed Plaintext Message 3*
Description	This test case aims at exploiting a vulnerability present in the 802.11i amendment allowing the Pairwise Transient Key (PTK) to be reinstalled during the four-way handshake using the TKIP protocol when the supplicant accepts delayed plaintext retransmissions of message 3.
Test scenario	In this test case, the access head acts as an AP and performs the 4-way handshake with the DUT up to message 3. Once message 3 is sent, it drops the message 4 sent by the DUT and waits for a few data frames to arrive. Then, it replays message 3 again and checks that the nonce used in the next data frame was NOT already used.
Expected behavior	DUT shall not reinstall the Pairwise Transient Key (PTK) when receiving the second message 3.
Success oracle	*Success if the DUT does not reuse previously used nonce*
Related weaknesses	*CWE-323: Reusing a Nonce, Key Pair in Encryption*
References	• IEEE Std 802.11™-2016 • CVE-2017-13077 • wpa_supplicant v2.3 • see https://github.com/kristate/krackinfo
DUT/SUT prerequisites	*DUT is in BSS station mode* *DUT supports IEEE 802.11i amendment*
Solutions and mitigations	Implement IEEE P802.11 countermeasures published on 2017/10/26 entitled “Addressing the Issue of Nonce Reuse in 802.11 Implementations”.

WIFI#2 - FCH_CKM: TKIP-PTK Reinstallation Attack / Consecutive Plaintext Message 3

FIELD	DESCRIPTION
Name	*TKIP-PTK Reinstallation Attack / Consecutive Plaintext Message 3*
Description	This test case aims at exploiting a vulnerability present in the 802.11i amendment allowing the Pairwise Transient Key (PTK) to be reinstalled during the four-way handshake using the TKIP protocol when the supplicant accepts consecutive plaintext retransmissions of message 3.
Test scenario	In this test case, the access head acts as an AP and performs the 4-way handshake with the DUT. Once message 2 is received, the access head sends two consecutive plaintext message 3 to the DUT. Then it checks that the nonce used in the next data frame was NOT already used.
Expected behavior	The DUT shall not reinstall the PTK when receiving the second message 3.
Success oracle	*Success if the DUT does not reuse previously used nonce.*
Related weaknesses	*CWE-323: Reusing a Nonce, Key Pair in Encryption*
References	• IEEE Std 802.11™-2016 • CVE-2017-13077 • wpa_supplicant v2.3 • see https://github.com/kristate/krackinfo
DUT/SUT prerequisites	*DUT is in BSS station mode.* *DUT supports IEEE 802.11i amendment.*
Solutions and mitigations	Implement IEEE P802.11 countermeasures published on 2017/10/26 entitled “Addressing the Issue of Nonce Reuse in 802.11 Implementations”.

WIFI#3- FCH_CKM: TKIP-PTK Reinstallation Attack / Consecutive Encrypted Message 3

FIELD	DESCRIPTION
Name	TKIP-PTK Reinstallation Attack / Consecutive Encrypted Message 3
Description	This test case aims at exploiting a vulnerability present in the 802.11i amendment allowing the Pairwise Transient Key (PTK) to be reinstalled during the four-way handshake using the TKIP protocol when the supplicant accepts consecutive encrypted retransmissions of message 3.
Test scenario	In this test case, the access head acts as an AP and performs the 4-way handshake with the DUT. Once message 2 is received, the access head sends two consecutive encrypted message 3 to the DUT. Then it checks that the nonce used in the next data frame was NOT already used.
Expected behavior	DUT shall not reinstall the Pairwise Transient Key (PTK) when receiving the second message 3.
Success oracle	*Success if the DUT does not reuse previously used nonce.*
Related weaknesses	*CWE-323: Reusing a Nonce, Key Pair in Encryption*
References	• IEEE Std 802.11™-2016 • CVE-2017-13077 • wpa_supplicant v2.3 • see https://github.com/kristate/krackinfo
DUT/SUT prerequisites	*DUT is in BSS station mode.* *DUT supports IEEE 802.11i amendment.*
Solutions and mitigations	Implement IEEE P802.11 countermeasures published on 2017/10/26 entitled “Addressing the Issue of Nonce Reuse in 802.11 Implementations”.

WIFI#4 - FCH_CKM: TKIP-PTK Reinstallation Attack / Plaintext and Encrypted Message 3

FIELD	DESCRIPTION
Name	TKIP-PTK Reinstallation Attack / Plaintext and Encrypted Message3
Description	This test case aims at exploiting a vulnerability present in the 802.11i amendment allowing the Pairwise Transient Key (PTK) to be reinstalled during the four-way handshake using the TKIP protocol when the supplicant accepts plaintext followed by encrypted retransmissions of message 3.
Test scenario	In this test case, the access head acts as an AP and performs the 4-way handshake with the DUT. Once message 2 is received, the access head sends a plaintext message 3 immediately followed by an encrypted message 3 to the DUT. Then it checks that the nonce used in the next data frame was NOT already used.
Expected behavior	The DUT shall not reinstall the Pairwise Transient Key (PTK) when receiving the second message 3.
Success oracle	*Success if the DUT does not reuse previously used nonce*
Related weaknesses	*CWE-323: Reusing a Nonce, Key Pair in Encryption*
References	• IEEE Std 802.11™-2016 • CVE-2017-13077 • wpa_supplicant v2.3 • see https://github.com/kristate/krackinfo
DUT/SUT prerequisites	*DUT is in BSS station mode.* *DUT supports IEEE 802.11i amendment*
Solutions and mitigations	Implement IEEE P802.11 countermeasures published on 2017/10/26 entitled “Addressing the Issue of Nonce Reuse in 802.11 Implementations”.

WIFI#5 - FCH_CKM: TKIP-GTK Reinstallation Attack in Group Key Handshake

FIELD	DESCRIPTION
Name	*TKIP-GTK Reinstallation Attack in Group Key Handshake*
Description	This test case aims at exploiting a vulnerability present in the 802.11i amendment allowing the Group Temporal Key (GTK) to be reinstalled during the group key handshake using the TKIP protocol
Test scenario	In this test case, the access head acts as an AP and performs the 4-way handshake first to install the GTK and then send ARP requests to increase the IV. Then, it reinstalls the GTK with IV=0 by sending a group message 1 and checks whether the DUT replies with a group message 2. Finally the access head replays the previous ARP request and checks that the DUT does NOT send an ARP response.
Expected behavior	The DUT shall not reinstall the Group Temporal Key (GTK) when receiving the group message 1.
Success oracle	*Success if the DUT does not reuse previously used nonce.*
Related weaknesses	*CWE-323: Reusing a Nonce, Key Pair in Encryption*
References	• IEEE Std 802.11™-2016 • CVE-2017-13080 • wpa_supplicant v2.3 • see https://github.com/kristate/krackinfo
DUT/SUT prerequisites	*DUT is in BSS station mode* *DUT supports IEEE 802.11i amendment.*
Solutions and mitigations	Implement IEEE P802.11 countermeasures published on 2017/10/26 entitled “Addressing the Issue of Nonce Reuse in 802.11 Implementations”.

WIFI#6 - FCH_CKM: TKIP-GTK Reinstallation Attack in the 4-way Handshake

FIELD	DESCRIPTION
Name	*TKIP-GTK Reinstallation Attack in the 4-way Handshake*
Description	This test case aims at exploiting a vulnerability present in the 802.11i amendment allowing the Group Temporal Key (GTK) to be reinstalled during the 4-way handshake using the TKIP protocol.
Test scenario	In this test case, the access head acts as an AP and performs the 4-way handshake first to install the GTK and then send ARP requests to increase the IV. Then, it reinstalls the GTK with IV=0 by sending again a message 3 and check whether the DUT replies with a message 4. Finally the access head replays the previous ARP request and checks that the DUT does NOT send an ARP response.
Expected behavior	The DUT shall not reinstall the GTK when receiving the second Message 3.
Success oracle	*Success if the DUT does not reuse previously used nonce.*
Related weaknesses	*CWE-323: Reusing a Nonce, Key Pair in Encryption*
References	• IEEE Std 802.11™-2016 • CVE-2017-13078 • wpa_supplicant v2.3 • see https://github.com/kristate/krackinfo
DUT/SUT prerequisites	*DUT is in BSS station mode.* *DUT supports IEEE 802.11i amendment*
Solutions and mitigations	Implement IEEE P802.11 countermeasures published on 2017/10/26 entitled “Addressing the Issue of Nonce Reuse in 802.11 Implementations”.

WIFI#7 - FCH_CKM: TKIP-IGTK Reinstallation Attack in Group Key Handshake

FIELD	DESCRIPTION
Name	*TKIP-IGTK Reinstallation Attack in Group Key Handshake*
Description	This test case aims at exploiting a vulnerability present in the 802.11w amendment allowing the Integrity Group Temporal Key (IGTK) to be reinstalled during the group key handshake using the TKIP protocol.
Test scenario	In this test case, the access head acts as an AP and performs the 4-way handshake first to install the PTK, GTK and IGTK followed by a DHCP handshake. Then, the access head waits for a probe request and reply with probe responses authenticated with the IGTK in order to increase the IGTK packet number (PN). The access head sends now a new group message 1 to reinstall the IGTK with PN=0. Finally the access head checks that the DUT does NOT reply with a message 2 (that would indicate that the IGTK was reinstalled).
Expected behavior	The DUT shall not reinstall the Integrity Group Temporal Key (IGTK) when receiving the group message 1.
Success oracle	*Success if the DUT does not reuse previously used nonce*
Related weaknesses	*CWE-323: Reusing a Nonce, Key Pair in Encryption*
References	• IEEE Std 802.11™-2016 • CVE-2017-13081 • wpa_supplicant v2.3 • see https://github.com/kristate/krackinfo
DUT/SUT prerequisites	*DUT is in BSS station mode* *DUT supports IEEE 802.11w amendment*
Solutions and mitigations	Implement IEEE P802.11 countermeasures published on 2017/10/26 entitled “Addressing the Issue of Nonce Reuse in 802.11 Implementations”.

WIFI#8 - FCH_CKM: TKIP-IGTK Reinstallation Attack in the 4-way Handshake

FIELD	DESCRIPTION
Name	*TKIP-IGTK Reinstallation Attack in the 4-way Handshake*
Description	This test case aims at exploiting a vulnerability present in the 802.11w amendment allowing the Integrity Group Temporal Key (IGTK) to be reinstalled during the 4-way handshake using the TKIP protocol.
Test scenario	In this test case, the access head acts as an AP and performs the 4-way handshake first to install the PTK, GTK and IGTK followed by a DHCP handshake. Then, the access head waits for a probe request and reply with probe responses authenticated with the IGTK in order to increase the IGTK packet number (PN). The access head sends now a new WPA message 3 to reinstall the IGTK with PN=0. Finally the access head checks that the DUT does NOT reply with a message 4 (that would indicate that the IGTK was reinstalled).
Expected behavior	The DUT shall not reinstall the IGTK when receiving the message 3.
Success oracle	*Success if the DUT does not reuse previously used nonce.*
Related weaknesses	*CWE-323: Reusing a Nonce, Key Pair in Encryption*
References	• IEEE Std 802.11™-2016 • CVE-2017-13079 • wpa_supplicant v2.3 • see https://github.com/kristate/krackinfo
DUT/SUT prerequisites	*DUT is in BSS station mode.* *DUT supports IEEE 802.11w amendment*
Solutions and mitigations	Implement IEEE P802.11 countermeasures published on 2017/10/26 entitled “Addressing the Issue of Nonce Reuse in 802.11 Implementations”.

WIFI#9 - FCH_COP: Check for Obsolete Protocols

FIELD	DESCRIPTION
Name	*Check for Obsolete Protocols*
Description	This test case aims at finding deprecated security protocols implemented in access points.
Test scenario	Listen for a beacon frame sent by the access point and check usage of obsolete protocols among WEP, TKIP, WPA1-PSK, PeerKey.
Expected behavior	WEP, TKIP, WPA1-PSK or PeerKey protocols are not used by the DUT.
Success oracle	Success if the access point does not support any of the obsolete protocols above.
Related weaknesses	*CWE-327: Use of a Broken or Risky Cryptographic Algorithm*
References	• *IEEE Std 802.11™-2016* • wpa_supplicant v2.3 • see https://github.com/kristate/krackinfo
DUT/SUT prerequisites	*DUT is in BSS access point mode.*
Solutions and mitigations	*Implement latest and secure cryptographic algorithms.*

WIFI#10 - FCH_COP: Group Message Replay

FIELD	DESCRIPTION
Name	*Group Message Replay*
Description	This test case consists in testing whether an STA accepts replayed group messages.
Test scenario	In this test case, the access head acts as an AP and performs the 4-way handshake with the DUT. Then, the access head sends a first ARP request, a second with a new nonce and finally replays the first ARP request. Then it checks that the DUT does NOT reply to the third ARP request.
Expected behavior	*The DUT shall not reply to replayed group messages.*
Success oracle	*Success if the DUT does not reply to the third ARP request.*
Related weaknesses	*CWE-323: Reusing a Nonce, Key Pair in Encryption*
References	• *IEEE Std 802.11™-2016* • wpa_supplicant v2.3 • see https://github.com/kristate/krackinfo
DUT/SUT prerequisites	*DUT is in BSS station mode* *DUT supports IEEE 802.11w amendment*
Solutions and mitigations	*Implement IEEE 802.11w replay check mechanism*

WIFI#11 - FCH_ IDENTIFICATION_SECURE _: Broadcast Deauthentication Attack

FIELD	DESCRIPTION
Name	*Broadcast Deautentication Attack*
Description	This test case aims at testing whether the STA accepts broadcast management frames that are not authenticated
Test scenario	Test case aims at testing whether the STA accepts broadcast management frames that are not authenticated. In this test case, the access head acts as an AP and performs the 4-way handshake first to install the PTK, GTK and more importantly IGTK. Then, the access head sends a broadcast deauthentication frame that is not authenticated with the BIP protocol. Then it checks that the DUT does NOT accept the packet and does NOT stop sending data frames.
Expected behavior	The DUT shall not accept non-authenticated deauthentication or disassociation frame when the IGTK is installed.
Success oracle	*Success if the DUT does not disconnect*
Related weaknesses	*CWE-306: Missing Authentication for Critical Function*
References	• *IEEE Std 802.11™-2016* • wpa_supplicant v2.3 • see https://github.com/kristate/krackinfo
DUT/SUT prerequisites	*DUT is in BSS station mode* *DUT supports IEEE 802.11w amendment.*
Solutions and mitigations	*Implement IEEE 802.11w authenticity check mechanism.*

ZIGBEE

Test Cases	
ZB#1	FCH_CKM : Key extraction using ZigBee Light Link development key.
ZB#2	FCH_CKM : Key extraction using ZigBee Light Link Certification key.
ZB#3	FCH_CKM : Key extraction using ZigBee Light Link leaked master key.
ZB#4	FCH_CKM : Key extraction using ZigBee default Trust Center link key.
ZB#5	FCH_CKM : Network key rotation after reset
ZB#6	FCH_CKM : Extract key from unencrypted OTA key provisioning
ZB#7	FCH_COP : ZigBee check legacy stack version
ZB#8	FCH_SECURE_CHANNEL_: Pairing requires physical interaction
ZB#9	FCH_CONFIGURATION_POLICY : Device answers to beacons requests
ZB#10	FCH_IDENTIFICATION_SECURE : ZLL Unicast reset to factory
ZB#11	FCH_IDENTIFICATION_SECURE : ZLL Broadcast reset to factory
ZB#12	FCH_IDENTIFICATION_SECURE : ZLL Unicast identification
ZB#13	FCH_IDENTIFICATION_SECURE : ZLL Broadcast identification

ZB#1 - FCH_CKM : Key extraction using ZigBee Light Link development key.

FIELD	DESCRIPTION
Name	*Key extraction using ZigBee Light Link development key.*
Description	The ZLL development key used prior to ZigBee Alliance certification phase is freely accessible in ZLL specifications: `"PhLi" || TrId || "CLSN" || RsID` The network key is transported encrypted during the RouterJoinRequest message (using response ID from ScanResponse command). The ZLL development key is reserved for development phases and must not be present in commercial devices This test case takes as input a PCAP containing the capture of the ZB commissioning phase. This test case will analyze the content of the given PCAP in order to ensure that the transport of the network key was not protected using the ZLL Development key.
Test scenario	Sniff a key provisioning and try to decipher network key in RouterJoinRequest messages with ZLL development key.
Expected behavior	while deciphering RouterJoinRequest message with ZLL development key, the network key must not be retrieved.
Success oracle	*Target device does not use compromised development key.*
Related weaknesses	*CWE-321: Use of Hard-coded Cryptographic Key*
References	*ZigBee Light Link Standard v1.0*
DUT/SUT prerequisites	The DUT is a ZigBee end device not connected to any ZigBee network. A ZigBee coordinator is ready to accept the DUT in its network *Target ZigBee Light Link devices.*
Solutions and mitigations	*Don't use development keys for production devices*

ZB#2 - FCH_CKM : Key extraction using ZigBee Light Link Certification key.

FIELD	DESCRIPTION
Name	*Key extraction using ZigBee Light Link Certification key*
Description	The ZLL certification key used during ZigBee Alliance certification phase is freely accessible in ZLL specifications: C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF The network key is transported encrypted during the RouterJoinRequest message (using response ID from ScanResponse command). The ZLL certification key is reserved for certification phases and must not be present in commercial devices. This test case takes as input a PCAP containing the capture of the ZB commissioning phase. This test case will analyze the content of the given PCAP in order to ensure that the transport of the network key was not protected using the ZLL Certification key.
Test scenario	Sniff a key provisioning and try to decipher network key in RouterJoinRequest messages with leaked certification key
Expected behavior	While deciphering RouterJoinRequest message with ZLL certification key, the network key must not be retrieved.
Success oracle	*Target device does not use compromised certification key.*
Related weaknesses	*CWE-321: Use of Hard-coded Cryptographic Key*
References	*ZigBee Light Link Standard v1.0*
DUT/SUT prerequisites	The DUT is a ZigBee end device not connected to any ZigBee network. A ZigBee coordinator is ready to accept the DUT in its network. *Target ZigBee Light Link devices.*
Solutions and mitigations	*Don't use certification keys for production devices.*

ZB#3 - FCH_CKM : Key extraction using ZigBee Light Link leaked master key.

FIELD	DESCRIPTION
Name	*Key extraction using ZigBee Light Link leaked master key*
Description	The ZLL master key used during key establishment has leaked: `9F5595f10257C8A469CBF42BC93FEE31` The network key is transported encrypted during the RouterJoinRequest message (using response ID from ScanResponse command). This test case takes as input a PCAP containing the capture of the ZB commissioning phase. This test case will analyze the content of the given PCAP in order to ensure that the transport of the network key was not protected using the leaked ZLL master key. This test case takes as input a PCAP containing the capture of the ZB commissioning phase. This test case will analyze the content of the given PCAP in order to ensure that the transport of the network key was not protected using the leaked ZLL master key.
Test scenario	Sniff a key provisioning and try to decipher network key in RouterJoinRequest messages with leaked master key.
Expected behavior	while deciphering RouterJoinRequest message with leaked ZLL master key, the network key must not be retrieved.
Success oracle	*Target device does not use compromised master key.*
Related weaknesses	*CWE-321: Use of Hard-coded Cryptographic Key;*
References	Twitter (https://twitter.com/MayaZigBee/status/582090997322149888)
DUT/SUT prerequisites	The DUT is a ZigBee end device not connected to any ZigBee network. A ZigBee coordinator is ready to accept the DUT in its network *Target ZigBee Light Link devices.*
Solutions and mitigations	*Stop using leaked cryptographic content.*

ZB#4 - FCH_CKM : Key extraction using ZigBee default Trust Center link key.

FIELD	DESCRIPTION
Name	*Key extraction using ZigBee default Trust Center link key*
Description	Attack on secrecy found by Zillner and Strobl [6] against ZigBee Light Link and Home Automation profiles. The HAPAP Profile states that: “The current network key shall be transported using the default TC link key in the case where the joining device is unknown or has no specific authorization associated with it.”. Default TC Link key : `0x5A 0x69 0x67 0x42 0x65 0x65 0x41 0x6C 0x6C 0x69 0x61 0x6E 0x63 0x65 0x30 0x39 ('ZigBeeAlliance09').` If an attacker is able to sniff a device join using the default TC link key, the active network key is compromised. Encryption is performed on the whole ZigBee frame. Test case takes as input a PCAP containing the capture of the ZB commissioning phase. This test case will analyze the content of the given PCAP in order to ensure that the transport of the network key was not protected using the default Trust Center link key.
Test scenario	Sniff a key provisionning and try to decrypt KeyTransportKey or KeyLoadKey commands with default TC key
Expected behavior	While deciphering KeyTransportKey or KeyLoadKey messages with default TC Link key, the network key must not be retrieved.
Success oracle	*Target device does not use default TC link key.*
Related weaknesses	*CWE-321: Use of Hard-coded Cryptographic Key;*
References	[6] Tobias Zillner and Sebastian Strobl. ZigBee exploited the good the bad and the ugly. 2015. ZigBee Home Automation Public Application Profile (2013.06)
DUT/SUT prerequisites	The DUT is a ZigBee end device not connected to any ZigBee network. A ZigBee coordinator is ready to accept the DUT in its network.
Solutions and mitigations	When interoperability between manufacturers is not required, the device shall use dedicated pre-configured link keys. (cf. ZigBee Home Automation Public Application Profile 5.3.3)

ZB#5 - FCH_CKM : Network key rotation after reset

FIELD	DESCRIPTION
Name	*Network key rotation after reset*
Description	Zillner and Strobl [6], point out that the network key shall be changed periodically in “a meaningful time period” or after “a certain number of messages”. This key rotation should also appear after a reset to factory. This test case takes as input: - the network key used before the reset, - a PCAP containing a capture of the commissioning phase. The PCAP will be analyzed to ensure that the network key used after the reset is different than the one used before.
Test scenario	Force a reset to factory, then sniff a new key provisioning then compare the two network keys.
Expected behavior	After a reset to factory, key rotation should be observed (the network key previously used should change)
Success oracle	*Network key should rotate.*
Related weaknesses	*CWE-323: Reusing a Nonce, Key Pair in Encryption*
References	[6] Tobias Zillner and Sebastian Strobl. ZigBee exploited: The good the bad and the ugly. 2015.
DUT/SUT prerequisites	*DUT paired with a [PSCAN-simulated] ZigBee coordinator*
Solutions and mitigations	*Implement key rotation mechanisms*

ZB#6 - FCH_CKM : Extract key from unencrypted OTA key provisioning

FIELD	DESCRIPTION
Name	*Extract key from unencrypted OTA key provisioning*
Description	Firstly realized by Joshua Wright [2]. Before ZigBeePro keys could only be pre-installed or OTA provisioned. When keys are OTA provisioned, they are sent in plaintext. This attack is always possible in ZigBeePro in standard security mode when a nonpreconfigured device joins a network : the TC sends the current network key unencrypted over-the-air. Test case takes as input a PCAP containing the capture of the ZB commissioning phase. Test case will analyze the content of the given PCAP in order to ensure that there's no key sent unencrypted.
Test scenario	*Sniff a key provisioning and look for plaintext keys*
Expected behavior	*Network key must not be retrieved in plaintext during an OTA key provisioning.*
Success oracle	*Target does not send/receive keys in plaintext*
Related weaknesses	*CWE-311: Missing Encryption of Sensitive Data*
References	[2] Joshua Wright. Killerbee: practical zigbee exploitation framework. 2009.
DUT/SUT prerequisites	The DUT is a ZigBee device ready to connect to an existing ZigBee Network.
Solutions and mitigations	*Do not use ZigBee Pro in standard security mode.*

ZB#7 - FCH_COP : ZigBee check legacy stack version

FIELD	DESCRIPTION
Name	*ZigBee check legacy stack version*
Description	ZigBee end device should not implement a legacy version (prior to ZigBee Pro).
Test scenario	*Send beacon request*
Expected behavior	If device answers beacon request, the implemented version shall be >= ZigBee Pro.
Success oracle	*Device should not implement a ZigBee stack prior to ZigBee Pro.*
Related weaknesses	*CWE-327: Use of a Broken or Risky Cryptographic Algorithm*
References	*CEA guideline*
DUT/SUT prerequisites	NA
Solutions and mitigations	*Best practices*

ZB#8 - FCH_SECURE_CHANNEL_: Pairing requires physical interaction

FIELD	DESCRIPTION
Name	*Pairing requires physical interaction*
Description	*Pairing ZigBee end device with its coordinator should imply physical interaction mechanism.*
Test scenario	*Pair with target, without any physical interaction*
Expected behavior	*A pairing **without** physical interaction should not happen*
Success oracle	*Pairing fails (physical interaction is required)*
Related weaknesses	*CWE-304: Missing Critical Step in Authentication*
References	*CEA guideline*
DUT/SUT prerequisites	The DUT is a ZigBee device ready to connect to an existing ZigBee Network.
Solutions and mitigations	*Implement a pairing requiring physical interaction.*

ZB#9 - FCH_CONFIGURATION_POLICY : Device answers to beacons requests

FIELD	DESCRIPTION
Name	*Device answers to beacon beacon*
Description	This test sends a 802.15.4 beacon broadcast request. According to specifications, the DUT is supposed to send a 802.15.4 beacon response containing information on its implementation (e.g., ZigBee stack version). Yet, such information may be used by an attacker to help hijacking the DUT
Test scenario	*Send beacon request*
Expected behavior	*Nothing, device should not reply with sensitive information*
Success oracle	*Device should not reply with sensitive information*
Related weaknesses	*CWE-200: Information exposure*
References	*CEA guideline*
DUT/SUT prerequisites	NA
Solutions and mitigations	-

ZB#10 - FCH_IDENTIFICATION_SECURE : ZLL Unicast reset to factory

FIELD	DESCRIPTION
Name	*ZLL Unicast reset to factory*
Description	DoS attack found by Ronen et al. [5] against ZigBee Light Link profile and tested on a Philips Hue. Consists in disconnecting the end device from network while impersonating controller
Test scenario	Send unicast reset to factory request with a transaction ID set to 0 while physically far away from DUT
Expected behavior	*Nothing: the frame should not be considered*
Success oracle	*Target does not disconnect from network*
Related weaknesses	*CWE-306: Missing Authentication for Critical Function*
References	[5] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. IoT goes nuclear: Creating a ZigBee chain reaction. 2017
DUT/SUT prerequisites	*DUT already connected on a ZB network*
Solutions and mitigations	*Critical function such as reset to factory must be authenticated*

ZB#11 - FCH_IDENTIFICATION_SECURE : ZLL Broadcast reset to factory

FIELD	DESCRIPTION
Name	*ZLL Broadcast reset to factory*
Description	DoS attack found by Ronen et al. [5] against ZigBee Light Link profile and tested on a Philips Hue. Consists in disconnecting the end device from network while impersonating controller
Test scenario	Send broadcast reset to factory request with a transaction ID set to 0 while physically far away from DUT
Expected behavior	*Nothing:* DUT should not consider the ZigBee interpan reset to factory frame
Success oracle	*Target does not disconnect from network*
Related weaknesses	*CWE-306: Missing Authentication for Critical Function*
References	[5] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. IoT goes nuclear: Creating a ZigBee chain reaction. 2017
DUT/SUT prerequisites	*DUT already connected on a ZB network*
Solutions and mitigations	Critical function such as reset to factory must be authenticated The end device should not take into account critical action sent in broadcast mode.

ZB#12 - FCH_IDENTIFICATION_SECURE : ZLL Unicast identification

FIELD	DESCRIPTION
Name	*ZLL Unicast identification*
Description	Test derived from the DoS attack found by Ronen et al. [5] against ZigBee Light Link profile.
Test scenario	Send unicast identification request with a transaction ID set to 0 while physically far away from DUT
Expected behavior	*Nothing:* DUT should not consider the ZigBee interpan identify frame.
Success oracle	*Target does identify*
Related weaknesses	*CWE-306: Missing Authentication for Critical Function*
References	[5] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. IoT goes nuclear: Creating a ZigBee chain reaction. 2017
DUT/SUT prerequisites	NA
Solutions and mitigations	*Identification should be authenticated*

ZB#13 - FCH_IDENTIFICATION_SECURE : ZLL Broadcast identification

FIELD	DESCRIPTION
Name	*ZLL Broadcast identification*
Description	Test derived from the DoS attack found by Ronen et al. [5] against ZigBee Light Link profile.
Test scenario	Send broadcast identification request with a transaction ID set to 0 while physically far away from DUT
Expected behavior	*Nothing: the frame should not be considered*
Success oracle	*Target does identify*
Related weaknesses	*CWE-306: Missing Authentication for Critical Function*
References	[5] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. IoT goes nuclear: Creating a ZigBee chain reaction. 2017
DUT/SUT prerequisites	NA
Solutions and mitigations	Identification should be authenticated Furthermore, from a functional point of view, broadcast identification does not make any sense.