BT: Possible to overwrite an existing bond during keys distribution phase when the identity address of the bond is known

Moderate **ceolin** published **GHSA-j76f-35mc-4h63** on Oct 5, 2021

Package

zephyr (west)

Affected versions

1.14.2, 2.4.0, 2.5.0

Patched versions

2.6.0

Description

During the distribution of the identity address information we don't check for an existing bond with the same identity address.This means that a duplicate entry will be created in RAM while the newest entry will overwrite the existing one in persistent storage.

Due to the duplicate entry in RAM the connection will succeed in pairing, but future reconnections will most likely pick the wrong keys entry, and re-encryption will end with MIC failure during encryption setup.Once the device reboots only the newest bond information will exist.The new bond storage will have the authentication and security level of the newest bond, i.e no security elevation.

Unable to resolve peer RPA and peer initiates new pairing procedure causes two entries in the keys storage with the same identity address.

This results in pairing failure when the bond information is selected on reconnection and wrong LTK causes disconnection with MIC error.

Reproduce

Local changes to imitate observed behavior, remote has cleared bond and is using a new IRK.

```
diff --git a/subsys/bluetooth/host/hci_core.c b/subsys/bluetooth/host/hci_core.c
index 836ea6807d..1f447d93b2 100644
--- a/subsys/bluetooth/host/hci_core.c
+++ b/subsys/bluetooth/host/hci_core.c
@@ -2594,10 +2594,13 @@ int bt_unpair(uint8_t id, const bt_addr_le_t *addr)
        if (IS_ENABLED(CONFIG_BT_SMP) &&
            (!addr || !bt_addr_le_cmp(addr, BT_ADDR_LE_ANY))) {
                bt_foreach_bond(id, unpair_remote, &id);
+               bt_rand(&bt_dev.irk[id], 16);
                return 0;
        }
        unpair(id, addr);
+
+       bt_rand(&bt_dev.irk[id], 16);
        return 0;
 }
```

Build shell with:

```
west build tests/bluetooth/shell/ -- -DCONFIG_BT_MAX_PAIRED=4
```

Step 1. Create a bond:
Peripheral shell

```
bt init
bt advertise on
bt oob
<copy oob addr>
<wait for connection>
bt security 2
```

Central shell

```
bt init
bt connect <oob addr>
<wait for pairing complete>
```

Step 2. Reconnect and re-pair with new IRK and RPA

Central shell

```
bt disconnect
bt clear all
bt oob
bt connect <oob addr>
```

Peripheral shell

```
<wait for connection>
bt security 2
bt bonds
<prints two entries with same identity address>
```

Step 3. Reconnect, peripheral will use wrong bond information.

Central shell

```
bt disconnect
bt connect <oob addr>
```

Peripheral shell

```
<wait for connection>
bt security 2
Security failed: F9:DC:CF:9C:89:87 (random) level 1 reason: Unspecified (8)
Disconnected: F9:DC:CF:9C:89:87 (random) (reason 0x3d)
```

Impact

What kind of vulnerability is it? Who is impacted?

Patches

- Fix on master: #33266 (2.6.0)
- Fix on 2.4: #33433 (unreleased)
- Fix on 2.5: #33432 (2.5.1-rc1)
- Fix on 1.14: #33718 (unreleased)

Workarounds

Is there a way for users to fix or remediate the vulnerability without upgrading?

References

Are there any links users can visit to find out more?

For more information

If you have any questions or comments about this advisory:

- Open an issue in example link to repo
- Email us at example email address

embargo: 2021-06-11
zepsec: ZEPSEC-138

Severity

Moderate **4.3** / 10

CVSS base metrics

Attack vector	Adjacent
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	None
Availability	Low

CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L

CVE ID

CVE-2021-3436

Weaknesses

CWE-694