Heap overflow vulnerability in the implementation of the gatt protocol

Moderate **Vudentz** published **GHSA-479m-xcq5-9g2q** on Nov 12, 2021

Package

BlueZ

Affected versions

5.58

Patched versions

None

Description

In my test, this does not require pairing to perform a 0click attack

I chose ubuntu for testing

In fact, I used two gatt vulnerabilities and a memory leak vulnerability on sdp to experiment

It is easy to implement a complete 0 click attack chain

First introduce the first gatt vulnerability. This vulnerability seems to be fixed in the new version of bluez, but it has not been fixed in ubuntu. I did not search for the cve number of this vulnerability, it seems to be an internal fix.

It is the out-of-bounds reading problem in cli_feat_read_cb

len = sizeof(state->cli_feat)-offset;
value = len? &state->cli_feat[offset]: NULL;

Unverified offset can cause the leakage of any address on the heap

I will focus on the second vulnerability
This is a heap overflow vulnerability caused by integer overflow
The calling path is prep_write_complete_cb->store_prep_data->append_prep_data

uint16_t len;
len = prep_data->length + length;
val = realloc(prep_data->value, len);
memcpy(val + prep_data->length, value, length);
prep_data->length = len;
The len here has an integer overflow vulnerability
The attacker can continue to write data to the port that allows prep write, len will become 0x0,0x200,...0xffff, and finally when 0xffff+0x100=0xff, the program executes realloc(,0xff)
But the following memcpy(val+0xffff, value, 0x100)

Caused a heap overflow
This can also cause a double free vulnerability through realloc(0)

With the above information leakage vulnerability, this can easily complete remote code execution in user mode

I choose a memory leak in sdp to try memory layout
I will report in the next report

It seems that there is no port on ubuntu in the initial state to allow prep write
So I used ordinary users to register a port through bluetoothctl in the test
The command is as follows
menu gatt
register-service 0xffff
register-characteristic 0xaaaa write
register-application

My poc may not trigger the vulnerability because of the wrong port handle, you can adjust it yourself after obtaining the corresponding handle through the findinfo message
I don't know if some devices have the service that allows prep write enabled by default
If it exists, this vulnerability will have a serious impact

The leak.c in the attachment is a leaked poc, you may need to change the handle to use cli_feat_read
double.c causes double free
Over.c causes heap overflow, which may not crash
The dump file is the scene when double free crashed

acknowlegment:ziming zhang of Ant Security Light-Year Lab

Best Regards,
ziming zhang

Severity

Moderate

CVE ID

CVE ID

No known CVE

Weaknesses

No CWEs