MWR LABS

Security Advisory

MediaTek GPU Extension Device Integer Overflow

11/05/2017

Software	MediaTek GPU Extension Device (GED)
Affected Versions	MediaTek 6735
Author	Mateusz Fruba
Severity	High
Vendor	MediaTek
Vendor Response	Fix Released

Description:

MediaTek is a company that provides system-on-chip solutions for wireless communications, HDTV, DVD and Blu-ray. A number of MediaTek clients including Huawei, and Neffos were found to be affected by a vulnerability in the MediaTek GPU Extension Device code.

The '/proc/ged' file implements an IOCTL interface vulnerable to an integer overflow. This vulnerability can be leveraged by local attackers to trigger a kernel heap memory corruption.

Impact:

Local attackers can exploit this issue to gain root privileges or achieve kernel mode code execution.

Cause:

This vulnerability is due to insufficient input validation of user supplied data.

Solution:

MediaTek clients can receive the security fix directly from the vendor.

Technical details

The MTK GPU Extension Device (GED) implementation can be found under '/drivers/misc/mediatek/gpu/ged/src' of the Android Open Source Project (AOSP). The GED code implements an IOCTL handler called 'ged_ioctl' which acts on data passed from user space to the kernel using the following structure:

```
typedef struct _GED_BRIDGE_PACKAGE
{
	unsigned int    ui32FunctionID;
	int             i32Size;
	void            *pvParamIn;
	int             i32InBufferSize;
	void            *pvParamOut;
	int             i32OutBufferSize;
} GED_BRIDGE_PACKAGE;
```

'I32InBufferSize' and 'i32OutBufferSize' are both user-controlled variables which are passed to the vulnerable 'ged_dispatch' function from 'ged_ioctl'. It should also be noted that signed integers are used for the size values. Firstly, 'ged_ioctl' initialises this structure by copying the data from user space using 'ged_copy_from_user' as shown below:

```
static long ged_ioctl(struct file *pFile, unsigned int ioctlCmd, unsigned long arg)
{
	int ret = -EFAULT;
	GED_BRIDGE_PACKAGE *psBridgePackageKM, *psBridgePackageUM =
(GED_BRIDGE_PACKAGE*)arg;
	GED_BRIDGE_PACKAGE sBridgePackageKM;
	...
		psBridgePackageKM = &sBridgePackageKM;
	if (0 != ged_copy_from_user(psBridgePackageKM, psBridgePackageUM,
sizeof(GED_BRIDGE_PACKAGE)))
	{
		GED_LOGE("Fail to ged_copy_from_user\n");
		goto unlock_and_return;
	}

	ret = ged_dispatch(psBridgePackageKM);
```

The ‘ged_dispatch’ function is then used to dispatch this data to the IOCTL as follows:

```
static long ged_dispatch(GED_BRIDGE_PACKAGE *psBridgePackageKM)
{
	int ret = -EFAULT;
	void *pvInt, *pvOut;
	typedef int (ged_bridge_func_type)(void*, void*);
	ged_bridge_func_type* pFunc = NULL;

	if ((psBridgePackageKM->i32InBufferSize >= 0) && (psBridgePackageKM-
>i32OutBufferSize >= 0) &&
		(psBridgePackageKM->i32InBufferSize + psBridgePackageKM->i32OutBufferSize <
GED_IOCTL_PARAM_BUF_SIZE))
	{
		pvInt = gvIOCTLParamBuf;
		pvOut = (void*)((char*)pvInt + (uintptr_t)psBridgePackageKM-
>i32InBufferSize);
		if (psBridgePackageKM->i32InBufferSize > 0)
		{
			if (0 != ged_copy_from_user(pvInt, psBridgePackageKM->pvParamIn,
psBridgePackageKM->i32InBufferSize))
				...
		}
```

The function checks whether ‘i32InBufferSize’ and ‘i32OutBufferSize’ are greater than or equal to 0. This is followed by another check to confirm that the sum of ‘i32InBufferSize’ and ’i32OutBufferSize’ does not exceed ‘GED_IOCTL_PARAM_BUF_SIZE’.

The code then attempts to copy this user supplied data into the ‘pvInt’ kernel memory buffer which is of fixed size ‘GED_IOCTL_PARAM_BUF_SIZE’ or 12KB.

The input validation on this function is insufficient and allows a malicious attacker to pass values which would lead to an integer overflow when calculating the sum of ‘i32InBufferSize’ and ‘i32OutBufferSize’.

If the following values were passed then a heap overflow will occur:

```
i32InBufferSize = GED_IOCTL_PARAM_BUF_SIZE + 0x1000

i32OutBufferSize = 2147483647
```

Detailed Timeline

Date	Summary
2016-10-22	Issue reported to MediaTek.
2016-11-16	MediaTek responded with confirmation of the issue.
2016-11-25	MWR queried MediaTek for the issue status and patch release plan.
2017-03-30	MWR queried MediaTek for the issue status and patch release plan.
2017-03-30	MediaTek confirmed that issue was fixed and a patch was available to its customers.