Talos Vulnerability Report

TALOS-2020-1112

NZXT CAM WinRing0x64 driver IRP 0x9c40a148 privilege escalation vulnerability

DECEMBER 16, 2020

CVE NUMBER

CVE-2020-13515

Summary

A privilege escalation vulnerability exists in the WinRing0x64 Driver IRP 0x9c40a148 functionality of NZXT CAM 4.8.0. A specially crafted I/O request packet (IRP) can cause an adversary to obtain elevated privileges. An attacker can send a malicious IRP to trigger this vulnerability.

Tested Versions

NZXT CAM 4.8.0

Product URLs

https://www.nzxt.com/camapp

CVSSv3 Score

8.8 - CVSS:3.0/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H

CWE

CWE-269 - Improper Privilege Management

Details

NZXT CAM is software designed as an all-in-one solution for computer hardware monitoring and performance. The software monitors fan speeds, CPU temperatures, network and RAM usage, as well as CPU/GPU frequencies for overclocking. It also has features for in-game overlays to track PC performance. The software also has an inventory for all devices that are installed on the PC at any given time.

The WinRing0x64 driver exists so that the NZXT CAM software can have access to the Windows Kernel as well as elevated privileges required to talk to PCI devices as well as making CPU/GPU configuration changes. This driver creates `\Device\WinRing0_1_2_0` that is accessible to any user on the system and this driver is used for all elevated tasks.

Using the IRP 0x9c402088 gives a low privilege user direct access to the `HalSetBusDataByOffset` function that is completely unrestrained. This allows a low privilege user to write data to the I/O bus, possibly changing PCI configuration information, or vendor specific data registers. This access could be used for privilege escalation.

```
00011298                  if (IoControlCode:0.d != 0x9c40a0c8 && (IoControlCode:0.d != 0x9c40a0d8 && (IoControlCode:0.d != 0x9c40a0dc &&
IoControlCode:0.d != 0x9c40a0e0)))
00011298                      if (IoControlCode:0.d == 0x9c40a108)
00011298                          goto label_11303
0001129f                      if (IoControlCode:0.d != 0x9c40a148)
0001129f                          goto completeRequest
000112a5                      uint64_t rbx_1 = zx.q(rdx->Type3InputBuffer:0.d)
000112ab                      if (rbx_1:0.d u< 8)
000112ab                          goto label_11306
000112ad                      int32_t* r9_1 = *(Irp + 0x18)
000112b1                      *rdi = 0
000112b4                      uint64_t rbx_2 = zx.q(rbx_1:0.d + 0xfffffff8)
000112b7                      uint64_t rcx_5 = zx.q(*r9_1)
000112e7                      var_18:0.d = *(r9_1 + 4)
000112f8                      rbx = zx.q(sbb.d(rbx_2:0.d, rbx_2:0.d, (HalSetBusDataByOffset(4, zx.q(zx.d(zx.q(rcx_5:0.d u>> 8):0.b)),
zx.q(((rcx_5:0.d u>> 3) & 0x1f) | ((rcx_5:0.d & 7) << 5)), r9_1 + 8, var_18, rbx_2:0.d):0.d - rbx_2:0.d) != 0) & 0xe0000003)
000112fe                      goto completeRequest
```

Credit

Discovered by Carl Hurd of Cisco Talos.

https://talosintelligence.com/vulnerability_reports/

Timeline

2020-07-17 - Vendor Disclosure
2020-08-10 - Vendor acknowledged; Talos issued copy of reports
2020-12-16 - Public Release

CREDIT

Discovered by Carl Hurd of Cisco Talos.

VULNERABILITY REPORTS

PREVIOUS REPORT NEXT REPORT

TALOS-2020-1111 TALOS-2020-1113