Talos Vulnerability Report

TALOS-2020-1110

NZXT CAM WinRing0x64 driver privileged I/O read IRPs information disclosure vulnerability

DECEMBER 16, 2020

CVE NUMBER

CVE-2020-13509, CVE-2020-13511

Summary

An information disclosure vulnerability exists in the WinRing0x64 Driver Privileged I/O Read IRPs functionality of NZXT CAM 4.8.0. A specially crafted I/O request packet (IRP) can cause the disclosure of sensitive information. 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

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

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.

CVE-2020-13509 - IRP 0x9c4060cc - IN Byte

Using the IRP 0x9c4060cc gives a low privilege user direct access to the `IN` instruction that is completely unrestrained at an elevated privilege level. This allows a low privilege user to read data from the processor I/O ports. This IRP reads only a single byte to the specific processor I/O port. This access could allow for information leakage of sensitive data.

```
0001118c              if (cond:1_1)
0001118c  label_1118c:
0001118c                  int32_t* rcx = *(Irp + 0x18)
00011190                  uint64_t r8_1 = zx.q(rdx->Type3InputBuffer:0.d)
00011194                  *rcx
0001123e                  if (IoControlCode:0.d == 0x9c4060cc)
0001123e                      unimplemented  {in al, dx}
0001123f                      *rcx = IoControlCode:0.b
00011241  _rbx0CompleteRequest:
00011241                      *rdi = r8_1:0.d
00011244                      goto rbx0CompleteRequest
```

CVE-2020-13510 - IRP 0x9c4060d0 - IN Word

Using the IRP 0x9c4060d0 gives a low privilege user direct access to the `IN` instruction that is completely unrestrained at an elevated privilege level. This allows a low privilege user to read data from the processor I/O ports. This IRP reads two bytes (one word) to the specific processor I/O port. This access could allow for information leakage of sensitive data.

```
00011237                  if (IoControlCode:0.d == 0x9c4060d0)
00011237                      unimplemented  {in ax, dx}
00011239                      *rcx = IoControlCode:0.w
0001123c                      goto _rbx0CompleteRequest
```

CVE-2020-13511 - IRP 0x9c4060d4 - IN Dword

Using the IRP 0x9c4060d4 gives a low privilege user direct access to the `IN` instruction that is completely unrestrained at an elevated privilege level. This allows a low privilege user to read data from the processor I/O ports. This IRP reads four bytes (one dword) to the specific processor I/O port. This access could allow for information leakage of sensitive data.

```
000111ae                  if (IoControlCode:0.d == 0x9c4060d4)
000111ae                      unimplemented  {in eax, dx}
000111af                      *rcx = IoControlCode:0.d
000111b1                      goto _rbx0CompleteRequest
```

Exploit Proof of Concept

This is an example of reading the first few processor I/O ports a DWORD at a time.

```
        [+] Getting Device Driver Handle
                [+] Device Name: \\.\WinRing0_1_2_0
                [+] Device Handle: 0x90
        [+] Setting Up Vulnerability Stage
                [+] Allocating Memory For Buffer
                        [+] Memory Allocated: 0x00000280C8B13F50
                        [+] Allocation Size: 0x10
                [+] Preparing Buffer Memory Layout
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
0000000F 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
01000000 00000000 00000000 00000000
01010000 00000000 00000000 00000000
0001010F 00000000 00000000 00000000
00000101 00000000 00000000 00000000...[output truncated]
```

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.
