SbieIniServer RunSbieCtrl Stack Overflow

High **DavidXanatos** published **GHSA-wpjw-jh2p-gwx7** 3 days ago

Package

sandboxie-plus/Sandboxie

Affected versions

<=1.17.2

Patched versions

1.17.3

Description

Summary

MSGID_SBIE_INI_RUN_SBIE_CTRL is intentionally handled before the normal sandbox and impersonation checks, and the handler accepts raw payload bytes after MSG_HEADER. For non-sandboxed callers, SbieIniServer::RunSbieCtrl forwards the entire trailing payload length to another overload that copies it into WCHAR ctrlCmd[128] with no bounds check. Any local interactive process can therefore smash the service stack by sending an oversized command payload.

Status: confirmed
Severity: high
CWE: CWE-121 (Stack-based Buffer Overflow)

Affected components:

- Sandboxie/core/svc/sbieiniserver.cpp
- - Sandboxie/core/svc/PipeServer.cpp

Details

Reachability

- PipeServer creates the service port with a NULL DACL, so any local process can connect.
- - SbieIniServer::Handler2 explicitly states that RUN_SBIE_CTRL is handled "from any process".

- The payload copy is only skipped for sandboxed callers; non-sandboxed low-privilege processes can still reach it directly.

Code references

- Sandboxie/core/svc/PipeServer.cpp:39
- Sandboxie/core/svc/PipeServer.cpp:241-257
- Sandboxie/core/svc/sbieiniserver.cpp:113-130
- Sandboxie/core/svc/sbieiniserver.cpp:1422-1427
- Sandboxie/core/svc/sbieiniserver.cpp:1445-1471

Technical detail

The first-stage handler forwards the entire trailing message body as a wide-character command:

```
if (ok)
{
    if (isSandboxed || msg->length <= sizeof(MSG_HEADER))
        status = RunSbieCtrl(hToken, NULL);
    else
        status = RunSbieCtrl(
            hToken,
            NULL,
            (WCHAR *)((UCHAR *)msg + sizeof(MSG_HEADER)),
            (msg->length - sizeof(MSG_HEADER)) / sizeof(WCHAR));
}
```

The second-stage helper then copies the attacker-controlled length into a fixed stack buffer:

```
WCHAR ctrlCmd[128] = { 0 };

...

memcpy(ctrlCmd, CtrlCmd, CtrlCmdLen * sizeof(WCHAR));
ctrlCmd[CtrlCmdLen] = L'\0';
```

There is no check that CtrlCmdLen <= 127.

PoC

1. Connect to the Sandboxie service port from a normal unsandboxed user process.
2. ii. Send MSGID_SBIE_INI_RUN_SBIE_CTRL.
3. iii. Set msg->length to sizeof(MSG_HEADER) + N * sizeof(WCHAR) with N > 127.
4. iv. Fill the trailing payload with controlled wide characters.
5. v. The service copies N characters into ctrlCmd[128] and overruns the stack.

Impact

- Crash of SbieSvc
- - Potential code execution as SYSTEM
- - Local privilege escalation candidate from any non-sandboxed user process in an interactive session

Recommended fix

- Define a dedicated request structure for MSGID_SBIE_INI_RUN_SBIE_CTRL with an explicit maximum field length.
- - Reject payloads longer than ARRAYSIZE(ctrlCmd) - 1.
- - Use bounded copies such as StringCchCopyNW.

Severity

High

CVE ID

CVE-2026-34461

Weaknesses

▶ CWE-121

Credits

Yanchon918s Reporter