NamedPipeServer OpenHandler Stack Overflow

High **DavidXanatos** published **GHSA-cf8x-f33g-vwfg** 3 days ago

Package

sandboxie-plus/Sandboxie

Affected versions

<= 1.17.3

Patched versions

1.17.3

Description

Summary

NamedPipeServer::OpenHandler trusts NAMED_PIPE_OPEN_REQ::server to be null-terminated even though the request parser only enforces a minimum packet size. Because PipeServer accepts variable-length messages, a sandboxed caller can append controlled data after NAMED_PIPE_OPEN_REQ and make wcscat(pipename, req->server) continue reading beyond the fixed server[48] field until it encounters an attacker-controlled terminator. This can overflow the stack buffer WCHAR pipename[160] inside the SYSTEM service.

Status: confirmed
Severity: high
CWE: CWE-121 (Stack-based Buffer Overflow), CWE-170 (Improper Null Termination)

Affected components:

- Sandboxie/core/svc/namedpipeserver.cpp
- - Sandboxie/core/svc/namedpipewire.h

Details

Reachability

- The pipe server is reachable by arbitrary clients because the server port is created with a NULL DACL.

- NamedPipeServer::Handler then restricts this specific message family to sandboxed callers, so this bug is reachable from inside the sandbox.
- The allowlist only validates req->name; req->server is copied without a length check.

Code references

- Sandboxie/core/svc/namedpipeserver.cpp:114-121
- Sandboxie/core/svc/namedpipeserver.cpp:164-197
- Sandboxie/core/svc/namedpipewire.h:54-60
- Sandboxie/core/svc/PipeServer.cpp:241-257

Technical detail

NAMED_PIPE_OPEN_REQ is defined as:

```
struct tagNAMED_PIPE_OPEN_REQ
{
    MSG_HEADER h;
    ULONG create_options;
    WCHAR name[64];
    WCHAR server[48];
};
```

OpenHandler only checks:

```
if (req->h.length < sizeof(NAMED_PIPE_OPEN_REQ))
    return SHORT_REPLY(STATUS_INVALID_PARAMETER);
```

After that it builds a path into a fixed stack buffer:

```
WCHAR pipename[160];

if (req->server[0]) {
    wcscpy(pipename, L"\\device\\mup\\");
    wcscat(pipename, req->server);
    wcscat(pipename, L"\\PIPE\\");
} else
    wcscpy(pipename, L"\\device\\namedpipe\\");
wcscat(pipename, req->name);
```

Because req->server is not length-checked or terminated, an attacker can send:

1. a valid allowed pipe name in req->name such as lsarpc,
2. ii. a non-zero server[0],
3. iii. no terminator inside the 48-wide-character server field,
4. iv. additional controlled wide characters after the struct,

5. v. a final terminator far enough away to overflow pipename.

PoC

1. Connect to the Sandboxie LPC/ALPC service port.
2. ii. Send MSGID_NAMED_PIPE_OPEN with h.length > sizeof(NAMED_PIPE_OPEN_REQ).
3. iii. Set name to L"lsarpc\0".
4. iv. Fill server[48] with non-zero wide characters and append an additional attacker-controlled wide-character tail after the struct.
5. v. Terminate the tail only after the total server string exceeds the remaining capacity of pipename[160].

Impact

- Crash of SbieSvc
- ◦ Potential code execution in the service context
- ◦ Practical sandbox escape / local privilege escalation candidate because the caller can be a sandboxed process and the vulnerable code runs in the privileged broker

Recommended fix

- Reject oversized MSGID_NAMED_PIPE_OPEN packets, or at minimum require explicit terminators inside name and server.
- ◦ Replace wcscpy / wcscat with bounded construction such as StringCchPrintfW.
- ◦ Validate the exact combined path length before building pipename.

Severity

High

CVE ID

CVE-2026-34464

Weaknesses

▶ CWE-121
▶ CWE-170

Credits

Yanchon918s Reporter