ProcessServer boxname Stack Overflows

High **DavidXanatos** published **GHSA-9cjg-vh9m-hhx4** 3 days ago

Package

sandboxie-plus/Sandboxie

Affected versions	Patched versions
<= 1.17.3	1.17.3

Description

Summary

Several ProcessServer handlers accept request structures with WCHAR boxname[34] and only enforce a minimum packet size. They then copy req->boxname with wcscpy into WCHAR[BOXNAME_COUNT] stack buffers, where BOXNAME_COUNT is 40. Because the service port accepts variable-length packets, an attacker can append controlled wide-character data after the struct and omit the terminator inside the fixed boxname[34] field. The broker then reads past the end of the request field and can overflow the destination stack buffer.

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

Affected components:

- Sandboxie/core/svc/ProcessServer.cpp
- - Sandboxie/core/svc/ProcessWire.h
- - Sandboxie/common/defines.h
  Confirmed sinks:
- KillAllHandler
- - SuspendAllHandler
- - RunSandboxedHandler (unsandboxed caller path)

Details

Code references

- Sandboxie/core/svc/ProcessWire.h:49-54
- Sandboxie/core/svc/ProcessWire.h:93-106
- Sandboxie/core/svc/ProcessWire.h:194-199
- Sandboxie/common/defines.h:54
- Sandboxie/core/svc/ProcessServer.cpp:262-268
- Sandboxie/core/svc/ProcessServer.cpp:567-574
- Sandboxie/core/svc/ProcessServer.cpp:2206-2212
- Sandboxie/core/svc/PipeServer.cpp:241-257

Technical detail

The request structures use a smaller fixed field:

```
struct tagPROCESS_KILL_ALL_REQ
{
    MSG_HEADER h;
    ULONG session_id;
    WCHAR boxname[34];
};

struct tagPROCESS_RUN_SANDBOXED_REQ
{
    MSG_HEADER h;
    WCHAR boxname[34];
    ...
};

struct tagPROCESS_SUSPEND_RESUME_ALL_REQ
{
    MSG_HEADER h;
    ULONG session_id;
    WCHAR boxname[34];
    BOOLEAN suspend;
};
```

The service copies them into larger stack arrays with no explicit termination check:

```
WCHAR TargetBoxName[BOXNAME_COUNT];
...
wcscpy(TargetBoxName, req->boxname);
```

and:

```
WCHAR boxname[BOXNAME_COUNT] = { 0 };
...
wcscpy(boxname, req->boxname);
```

Since PipeServer permits packets larger than the request struct, an attacker can:

1. send a packet with h.length > sizeof(request),
2. ii. fill boxname[34] with non-zero data,
3. iii. append additional controlled wide characters after the struct,
4. iv. place the terminator only after more than BOXNAME_COUNT - 1 characters total.
   wcscpy then walks into the attacker-controlled tail and overruns the stack destination.

Reachability

- The broker port is created with a NULL DACL, so any local process can connect.
- - ProcessServer::Handler dispatches these message IDs without first impersonating or rejecting untrusted callers.
- - In KillAllHandler and SuspendAllHandler, the unsafe copy happens before the authorization checks.

PoC

1. Connect to the Sandboxie service port.
2. ii. Send either MSGID_PROCESS_KILL_ALL or MSGID_PROCESS_SUSPEND_RESUME_ALL with an oversized packet.
3. iii. Make the in-struct boxname[34] non-terminated.
4. iv. Append a controlled wide-character tail after the struct.
5. v. Place the final terminator only after the combined string length exceeds 39 characters.

Impact

- Crash of SbieSvc
- - Potential code execution as SYSTEM
- - Local privilege escalation candidate from an unprivileged local process

Public overlap check

- The public GitHub Security overview for sandboxie-plus/Sandboxie was checked on 2026-03-12.
- - No public advisory title matched this ProcessServer boxname parsing issue.

Recommended fix

- Require an in-struct terminator for fixed-length string fields before using them.
- - Prefer bounded copies such as StringCchCopyNW.

- Consider rejecting oversized packets for fixed-layout request structures where trailing data is not expected.

Severity

High

CVE ID

CVE-2026-34462

Weaknesses

- CWE-121
- CWE-170

Credits

Yanchon918s Reporter