



# Nitro PDF Pro mailDoc() Use-After-Free Vulnerability

[← Back to Advisories](#)

## CVE NUMBER

**CVE-2025-69627**

## VENDOR

**Nitro**

## CREDIT

**Abu**

## Description

A use-after-free vulnerability exists in Nitro PDF Pro 14.41.1.4 during the processing of the JavaScript method `this.mailDoc()`. When `mailDoc()` is invoked via JavaScript embedded in a crafted PDF document, Nitro PDF Pro frees an internal object but continues to use the dangling pointer in subsequent UI or string handling routines. This results in a use-after-free condition, allowing the application to be crashed by simply opening the malicious document.

## Details

The issue occurs because an internal `XID` object created during the handling of `this.mailDoc()` is freed prematurely, while the same pointer continues to be passed to subsequent functions.

The vulnerable flow can be simplified as follows:

```
v11 = sub_2B6C0(a1, 0); // Allocate XID object
...
CMFCRibbonInfo::XID::XID(v11);
j_j_free_0(v11); // Free XID object

v20 = CStringT(..., L"document.mailDoc()");
v21 = sub_DE50(...);

// Use-After-Free
v22 = CDumpContext::operator<<(v11);

if (CRichEditView::XRichEditOleCallback::ContextSensit
    (CRichEditView::XRichEditOleCallback *)v27,
    v22))
{
    ...
}
```

In other words, the internal object `v11` is freed via `j_j_free_0(v11)`, but continues to be used in `CDumpContext::operator<<` and subsequent UI/string handling paths. During this process, residual or reallocated heap data associated with the freed memory may be passed into string comparison routines, leading to undefined behavior.

In some execution environments, the dangling pointer immediately references an invalid address, resulting in an access violation. In other cases, the pointer propagates further into string comparison routines such as `wcscmp` before triggering a crash. While the exact crash behavior may vary depending on the heap state, the root cause remains the same: a use-after-free condition.

This vulnerability can be triggered by simply invoking `this.mailDoc();` without any arguments.

## Timeline

**2025-12-10** - Vulnerability reported to Vendor

**2026-02-02** - Vendor Patch Release

**2026-04-06** - Public Release

© 2026 JeroScope