Essential Eight on Linux, Part 6 of 8: Restrict Microsoft Office Macros on Ubuntu 26.04 LTS

May 1, 2026 / Gavin Jackson

essential-eight asd ism ubuntu linux libreoffice apparmor security defender-for-office-365

This is the Essential Eight mitigation that maps least cleanly to Linux.

To be honest, the temptation to mark this one as N/A and move on with my life was very real. But the more I thought about it, the more I wanted to explore how office suites, including open-source alternatives, are actually used from Linux and what threat vectors they really introduce.

Ubuntu 26.04 LTS does not have a neat, native equivalent to the Windows-centric control set around Microsoft Office desktop macros. Most Linux desktops are using **LibreOffice**, **Office on the web**, or some form of remote Windows application delivery if they still depend on the full Microsoft Office desktop stack.

So if you want a defensible Linux implementation, the right move is not to pretend the platforms are the same. The right move is to translate the security intent.

What ASD is trying to achieve

The macro mitigation exists because document-borne code execution has been one of the most successful intrusion paths for years.

The underlying security objective is:

- do not let untrusted documents execute code
- limit macro use to business-justified cases
- prefer trusted locations, trusted publishers, and stronger document handling

That objective still matters on Linux, even if the exact technology stack changes.

Ubuntu 26.04 LTS reference implementation

Resolute Raccoon highlights

Resolute Raccoon does not introduce a magical Linux answer for Microsoft Office macros, but it does improve the surrounding control surface:

- **LibreOffice 25.8** is a meaningful update from the 24.04 baseline
- the desktop permission model is more visible through the **Security Center**
- snap and portal permissions are easier to reason about when document workflows cross trust boundaries

That still leaves this mitigation as one of the least direct Linux mappings, but the platform is a bit easier to harden thoughtfully.

1. Prefer web-based productivity and trusted SaaS workflows

If a user does not need rich desktop macro capability, do not give it to them.

For many Ubuntu users, the safest path is:

- Microsoft 365 on the web
- LibreOffice with macros disabled or heavily restricted
- PDF output for distribution

That immediately removes a large amount of attack surface.

2. Harden LibreOffice macro behaviour

LibreOffice is not Microsoft Office, but it still supports macros and scripting. On Ubuntu 26.04, the practical control set is:

- set macro security to high or very high
- restrict trusted file locations
- avoid enabling macros from email downloads or user-writable locations
- disable unneeded scripting and Java dependencies where practical
- confine LibreOffice with AppArmor if handling higher-risk content

If only a small number of teams genuinely need macros, make that an exception workflow rather than the desktop default.

3. Treat downloads as hostile until proven otherwise

For Linux desktops, a lot of the real defence is upstream of the office suite:

- email attachment scanning
- content disarm and reconstruction
- remote browser isolation
- controlled file transfer from less trusted domains

By the time the document reaches LibreOffice, you want as much hostile content stripped or quarantined as possible.

4. Use AppArmor and mount options as blast-radius controls

AppArmor will not magically turn LibreOffice into a secure macro runtime, but it can reduce the damage a compromised document handler can do.

Combine that with:

- `noexec` on download-heavy locations where feasible
- least-privilege user accounts
- restricted local admin
- strong browser and email client hardening

and the attack path becomes much less forgiving.

ISM control mapping

The October 2024 Essential Eight to ISM mapping links this mitigation to these controls:

ISM control	Linux interpretation on Ubuntu 26.04 LTS
`ISM-1671`	Restrict macro execution in office productivity software and allow only approved business use cases.
`ISM-1488`	Limit macro execution to trusted locations, trusted publishers, or specific approved workflows where possible.
`ISM-1672`	Prevent execution of macros from untrusted or user-controlled sources such as email downloads.
`ISM-1673`	Apply stronger restrictions at higher maturity levels so macro use becomes the exception rather than the rule.
`ISM-1489`	Reduce or eliminate unnecessary macro capability on user systems.

Where native parity does not exist

This is the key Linux gap:

- no direct Ubuntu equivalent to Group Policy settings for the Microsoft Office desktop stack
- no exact analogue for Windows-specific VBA controls such as blocking Win32 API access from Office macros
- no clean one-to-one mapping for organisations that still rely on heavy VBA desktop automation

If your business still depends on dense Microsoft Office macro workflows, Ubuntu is probably not the host platform on which you want to solve that problem.

> ***Watch this space: France, sovereignty, and what comes after LibreOffice***
>
> *One development I think is worth watching closely is the French Government's broader move toward a sovereign, open-source desktop and collaboration stack. In April 2026, DINUM directed ministries to plan for reducing dependence on proprietary operating systems and non-European digital platforms, with each ministry expected to formalise its approach by autumn. That matters here because once a government starts pulling at the Windows and cloud-services thread, Microsoft Office dependency quickly becomes part of the same conversation.*

> *France has history in this area. Large public-sector migrations to OpenOffice.org and later LibreOffice have already shown that office-suite change is possible at scale when the rollout is phased, politically backed, and supported properly. But ministry-wide migration is still much harder than a single-agency success story. The real friction is not opening a `.docx` file. It is the long tail of macros, templates, line-of-business integrations, digital signature workflows, and years of institutional habit built around Word and Excel.*
>
> *What feels different this time is the broader sovereignty agenda around **La Suite**, France's state-backed open and sovereign workspace, and the parallel emergence of **Euro-Office** in the wider European ecosystem. I do not think Euro-Office is something to treat as mature or settled yet, but it is interesting because it suggests the conversation may move beyond "just use LibreOffice" toward a new generation of open, Europe-governed office tooling with a stronger focus on Microsoft format compatibility.*
>
> *For anyone thinking about Linux adoption in government or regulated sectors, that could have global implications. If France and other European public bodies can create real momentum behind open formats, sovereign collaboration tools, and credible alternatives to entrenched Office workflows, the Linux story around productivity software may look very different a few years from now. For now, I would treat this as a genuine **watch this space** topic.*

> ***How much LibreOffice or OpenOffice macro exploitation have we actually seen?***
>
> *The honest answer is: far less than the long history of Microsoft Office macro abuse, at least in the public record. I have not found strong evidence of widespread, modern LibreOffice macro campaigns on the same scale as the classic Word or Excel malware ecosystem. That said, I do not think the risk should be dismissed.*
>
> *There are two reasons for that. First, LibreOffice has had real security issues around macro execution and warning bypasses, including **CVE-2019-9853** and **CVE-2023-6186**, both of which affected how macro-related actions could bypass normal user protections. Second, there is historical evidence that attackers and researchers have at least experimented with cross-platform OpenOffice style macro malware, such as the older **BadBunny** proof-of-concept worm.*
>
> *So my takeaway is not "LibreOffice macros are a major in-the-wild epidemic." It is more that the macro attack path still exists, the security controls around it have needed fixing over time, and Linux environments should avoid a false sense of safety just because the volume of public exploitation appears lower than on Windows.*

> ***What about running Microsoft Office under Wine or similar compatibility layers?***

> *Yes, Office emulation on Linux is still a thing in the broad sense. People continue to try **Wine** and commercial compatibility layers such as **CrossOver** when they need Microsoft Office without a full Windows VM. But I would treat that as a compatibility workaround, not a strong security answer.*
>
> *In fact, from a security point of view it can make things worse. If you run a Windows Office stack on Linux through Wine or a similar layer, you may reintroduce the very macro and document attack surface you were trying to avoid, while also adding patching, support, and visibility problems. It can become harder to reason about what is actually supported, how it is updated, and how well Linux-native controls such as AppArmor, browser isolation, or standard package governance really contain the resulting workflow.*
>
> *There is also a practical warning sign here: CodeWeavers announced in January 2026 that CrossOver would stop its modest support for newer Microsoft Office 365 and Copilot 365 workflows. That does not mean Wine-style approaches disappear, but it does reinforce the point that emulation is not the same thing as a clean, supportable desktop standard for secure environments.*

Linux-friendly commercial alternatives

When native parity does not exist, I would look at controls that are endpoint-agnostic or Linux-compatible:

- **Microsoft Defender for Office 365** for email attachment detonation, Safe Attachments, and broader document threat reduction before content reaches the Linux endpoint
- **OPSWAT MetaDefender Deep CDR** for content disarm and reconstruction on inbound files
- **Cloudflare Remote Browser Isolation** or **Menlo Security** where risky document access or web-delivered content should be opened in an isolated session instead of directly on the endpoint

These are not all "macro control" products in the strict Windows sense, but they are commercially mature ways to reduce the same attack path while still supporting Linux users.

Good compensating controls

If you need to stay Ubuntu-first and keep the implementation practical, I would combine:

- LibreOffice macro security set high or very high
- no default macro enablement for standard users
- document sanitisation at mail or file ingress
- browser isolation for high-risk content sources
- AppArmor confinement for document handling applications
- Office on the web instead of local rich clients where feasible

That is the honest Linux answer.

The bottom line

This mitigation does not translate perfectly to Ubuntu 26.04 LTS, and pretending otherwise leads to bad architecture.

The right Linux implementation is to preserve the **security intent**: restrict document-borne code execution, minimise local macro use, sanitise risky content early, and isolate the few workflows that genuinely require more power. Where the native stack stops short, commercial gateway and isolation controls are the cleanest answer.

References

- ASD Essential Eight maturity model and ISM mapping (October 2024)
- LibreOffice macro security help
- AppArmor on Ubuntu
- Microsoft Defender for Office 365
- OPSWAT Deep CDR technology
- Cloudflare Remote Browser Isolation
- France phases out proprietary operating systems on workstations
- La Suite
- La Suite Docs
- Euro-Office launch announcement
- LibreOffice security advisory for CVE-2019-9853
- NVD entry for CVE-2023-6186
- BadBunny OpenOffice macro worm coverage
- CodeWeavers on ending support for newer Microsoft Office 365 in CrossOver