Essential Eight on Linux, Part 5 of 8: Multi-Factor Authentication on Ubuntu 26.04 LTS

April 30, 2026 / Gavin Jackson

essential-eight asd ism ubuntu linux teleport entra-id active-directory mfa security

Multi-factor authentication on Linux is easy to do badly.

There are plenty of ways to bolt a second factor onto a PAM stack and declare victory. That may technically satisfy a checklist, but it does not always produce a manageable, phishing-resistant, or scalable control.

For Ubuntu 26.04 LTS, I think the strongest reference model is:

- central identity provider
- strong MFA at the identity layer
- brokered access to Linux resources
- short-lived credentials instead of static secrets

That is why Teleport keeps showing up in this series.

What ASD is trying to achieve

The point of MFA is not just to add another prompt. It is to make stolen passwords materially less useful and to raise the cost of account compromise, especially for privileged and externally accessible services.

For Linux resources, the main targets are:

- SSH access
- web administration portals
- privileged access workflows
- remote access into production or sensitive environments

Ubuntu 26.04 LTS reference implementation

Resolute Raccoon highlights

This is one area where Ubuntu 26.04 LTS clearly moved the platform forward. Canonical now ships **authd** in the official repositories, with support for **Microsoft Entra ID**, **Google IAM**, and generic **OIDC** providers.

That means Ubuntu now has a much cleaner first-party path for cloud-backed identity and Linux login than it did on Ubuntu 24.04 LTS. I still prefer Teleport for privileged access brokering and stronger JIT control, but authd makes the identity layer underneath that design more compelling.

1. Put MFA in front of access, not just on the host

Ubuntu supports host-level MFA through PAM modules, and there are cases where that is useful. But at fleet scale, the cleaner pattern is to put MFA at the identity and access broker layer.

That gives you:

- central policy
- stronger reporting
- consistent user experience
- easier enforcement of phishing-resistant methods

2. Use Teleport for SSH and privileged Linux access

Teleport is a strong fit for Linux MFA because it can enforce MFA before issuing the short-lived certificate used to access the target resource.

That matters because it moves you away from:

- static SSH keys
- password-based SSH
- inconsistent host-by-host MFA configuration

and toward:

- SSO
- MFA at login and re-auth points
- certificate-based access
- session recording and audit

3. Use Entra ID for modern SSO and MFA policy

If your organisation already uses Microsoft 365, Entra ID is often the most natural place to enforce:

- MFA requirements
- conditional access
- device or risk-based access policies
- phishing-resistant methods such as FIDO2

Teleport integrates well here because it can consume Entra as the identity provider while still presenting a Linux-native access pattern to the target hosts.

4. Keep Active Directory where it still makes operational sense

On some internal networks, Ubuntu joined to AD through SSSD still makes sense for identity and local account mapping.

That is not in conflict with Entra-backed MFA. You can use AD for traditional enterprise integration and Teleport plus Entra for the stronger external access and privileged access story.

5. Prefer phishing-resistant MFA where possible

> ***What makes MFA phishing resistant?***
>
> *This was actually a new term for me when I started digging into the detailed ISM controls rather than just the high-level Essential Eight wording.*
>
> *In simple terms,* ***phishing-resistant MFA*** *means the second factor is designed so it is much harder for an attacker to steal, replay, or trick a user into handing it over through a fake login page. That is the big difference between something like a* ***FIDO2 security key or passkey*** *and a weaker method such as* ***SMS codes*** *or even app-based one-time passwords.*
>
> *If a user can be convinced to type the code into a fake website, approve a rogue prompt, or hand the factor to an attacker in real time, it is not very phishing resistant. By contrast, modern public-key methods bind the authentication to the legitimate service, which makes credential theft through fake sign-in pages much harder.*
>
> *That is why ASD keeps pushing the concept: not all MFA is equal, and some forms of MFA hold up much better when someone is actively trying to phish your users.*

For sensitive Linux administration, my preference order is usually:

1. FIDO2 or WebAuthn security keys
2. smart cards or certificate-backed methods
3. app-based OTP as a fallback
4. avoid SMS unless legacy realities force it

This matters even more for privileged access. If the user can reach production Linux with the identity, the MFA method should not be the weakest thing in the chain.

ISM control mapping

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

ISM control	Linux implementation on Ubuntu 26.04 LTS
ISM-1504	Require MFA for users of important systems and services that expose Linux administration or sensitive data.
ISM-1679	Enforce MFA for externally accessible services, including brokered SSH and administrative web portals.
ISM-1680	Apply MFA consistently to administrative access paths rather than leaving privileged exceptions.
ISM-1892	Use stronger MFA controls for higher-risk access scenarios and sensitive systems.
ISM-1893	Prefer phishing-resistant MFA methods where practical for privileged or exposed services.
ISM-1681	Ensure MFA is integrated into remote access and privileged workflows, not left as an optional extra.
ISM-1401	Protect authentication events with logging and auditability.
ISM-1173	Review MFA scope and user coverage regularly.
ISM-0974	Protect credentials and authentication secrets from misuse or reuse.
ISM-1872	Extend MFA coverage to important applications and administrative interfaces.
ISM-1873	Apply MFA to privileged and higher-risk access paths with stronger assurance.
ISM-1682	Avoid weak or inconsistent second-factor implementations across the environment.
ISM-1683	Revalidate MFA configuration, exemptions, and method strength over time.
ISM-1815	Apply higher-maturity authentication controls across privileged workflows.
ISM-1906	Align stronger authentication requirements with other uplift controls.
ISM-1228	Monitor authentication activity for misuse, anomalies, and incident response.
ISM-0123	Protect security logs related to authentication.
ISM-0140	Retain sufficient authentication records for investigations.
ISM-1819	Improve authentication assurance over time instead of freezing the design at rollout.
ISM-1505	Protect remote access paths with strong authentication.
ISM-1874	Enforce stronger MFA for high-risk remote and administrative use cases.
ISM-1894	Prefer phishing-resistant MFA in higher maturity implementations.
ISM-1907	Reduce reliance on weaker factors as the environment matures.
ISM-0109	Protect authentication systems and their supporting infrastructure.

Where native Linux can get messy

PAM-based MFA works, but on its own it can become:

- inconsistent across hosts

- hard to audit centrally
- brittle during break-glass events
- uneven across SSH, web apps, and administrative tools

That is why I prefer identity-brokered MFA over host-by-host MFA as the primary pattern.

Practical Linux pattern

For an Ubuntu 26.04 estate, I would aim for:

- no password-based SSH for administrators
- Teleport in front of privileged Linux access
- Entra ID as the SSO and MFA authority where appropriate
- AD integration on hosts where traditional enterprise identity is still needed
- FIDO2 security keys for privileged users
- local PAM MFA only for targeted use cases or resilience paths

That gives you a control that scales and remains auditable.

The bottom line

Ubuntu 26.04 LTS can absolutely meet the intent of the Essential Eight MFA mitigation, but the strongest implementation is not "turn on another PAM module everywhere and hope for the best."

The better answer is **central identity**, **strong MFA**, and **brokered Linux access** using tools like **Teleport**, with **Entra ID** or another mature IdP enforcing the real authentication policy. Resolute Raccoon's new **authd** packaging is one of the more useful Linux identity improvements in this release because it gives Ubuntu a more official bridge into modern cloud identity.

References

- ASD Essential Eight maturity model and ISM mapping (October 2024)
- Implementing multi-factor authentication
- Guidelines for system hardening
- Teleport authentication and MFA
- Teleport Microsoft Entra ID SSO
- Ubuntu Active Directory integration
- Microsoft Entra ID documentation
- authd documentation