Security

What NetWare taught us about AI agent permissions.

In the 1990s we learned that what someone was granted and what they could actually do were rarely the same thing. Thirty years later, AI agents have made that lesson urgent again.

Toltec engineering | 25 September 2026 | 3 minute read

The short version

- An agent's real reach is its effective permissions, not the list on its app registration.
- Effective permissions come from several places at once: direct grants, inheritance, groups and the identity it runs as.
- Work out the whole path before go-live, and again after every change.
- Stop inheritance where it is not needed, and grant at the narrowest point that works.
- Compare granted access with observed use, check infrequent tasks, and remove permissions that are no longer required.

Effective rights, then

On a NetWare file server, access to a folder rarely came from one place. A user might hold a trustee assignment on the folder itself, inherit rights granted higher up the tree, receive more through group membership, and gain everything a colleague had through security equivalence. The rights that mattered were the effective rights: the sum of all of them, less anything an inherited rights filter blocked.

Most of the access problems we untangled in that era had the same shape. Nobody had granted the wrong thing on purpose. Several reasonable grants, made years apart by different people, added up to something nobody intended.

Effective permissions, now

An AI agent in Microsoft 365 brings the same need to trace effective access, at far greater speed. Its reach depends on its authentication mode, consented application permissions[1] or delegated scopes, and the controls on the resource it accesses. Follow each identity and token separately: permission used by one connection does not automatically apply to another. With delegated access, the agent is limited by both its scopes and the person's rights; broad groups or sharing can still widen the content it can reach.

The difference is that an agent acts on what it can reach, quickly and at scale, and it can be instructed by the content it reads. A permission nobody noticed for years becomes a path someone else can use in minutes.

Work out the path, not the grant

- List every source of access: application permissions, delegated scopes, directory roles, group memberships and sharing on the content itself.
- Trace the effective result for the data the agent will touch, not just the permissions on its registration.
- Grant at the narrowest point that works: a named site rather than every site[2], a scoped mailbox rather than every mailbox[3].
- Stop inheritance where it is not needed. A dedicated site or library for agent work is often simpler than filtering a busy one.
- Keep the agent's identity separate from people's accounts, so its effective permissions can be reasoned about on their own.

Measure what it actually uses

Assigned permissions describe potential access. Sign-in and audit logs provide evidence of use, subject to their coverage and retention. Compare the two, check infrequent tasks and recovery needs, then remove permissions that are no longer required. The same review discipline helped keep file server rights under control thirty years ago.

What we would do differently

Start with the data, not the agent. Decide which sites, mailboxes and records the job genuinely needs, then build the narrowest identity that reaches exactly those. Working backwards from the data makes the effective permissions easier to test and explain.

References

1. Microsoft Graph permissions reference Microsoft Learn
2. Microsoft Graph permissions reference: Sites.Selected Microsoft Learn
3. Role Based Access Control for Applications in Exchange Online Microsoft Learn

Planning an agent that needs access to your data?

Talk to an engineer

Read the current edition: toltec.io/note-netware-agent-permissions.html