FOXT.

Access Control Excellence

Privileged Account Access Management: Why Sudo Is No Longer Enough

The new privileged access management solutions available on the market today provide highly efficient and effective alternatives to sudo. Using these modern approaches, you will be able to reduce the risk of insider fraud, streamline regulatory compliance, and greatly reduce the effort required to administer your server estates.

Sudo is a free open-source access control tool that requires costly and labor-intensive custom configuration to meet privilege access management and compliance requirements.

Sudo may be a reasonable solution for controlling privileged accounts for organizations with small server infrastructures, but it lacks the administration efficiencies, architectural vision, and security-related compliance requirements that are needed to effectively protect critical assets for organizations running hundreds and thousands of diverse Unix and Linux servers. There are six primary challenges with using sudo to control privileged accounts:

First, sudo lacks efficient, centralized administration. System administrators typically spend hours and hours building and distributing sudoers files across the server estates. Sudo lacks the ability to easily put servers into local categories, classify users by various roles, and define the associated access rules, methods readily available through the latest privileged access management solutions on the market.

Second, let's talk about the security risk. Sudo is controlled by local files, and the burden is on the security admins to distribute these files appropriately. To do the distribution well, each server typically needs to have a unique file, but in most cases, shortcuts are taken by administrators, which results in giving administrative users too much privilege. As well, the sudo configuration file is stored in a way that local administrators could easily make modifications….a big security risk.

Third, sudo creates a compliance issue. The distributed sudo conf files are not liked by auditors because they utilize "static trust". Organizations using sudo may have problems passing audits.

Fourth, organizations using sudo must utilize a custom-built distribution method, and that custom method must be maintained, resulting in hidden costs.

Fifth, sudo does not inherently provide the ability to holistically link multi-factor, contextual authentication as part of the privileged user authorization process. Contextual authentication provides greater flexibility in the methods used to authenticate any given user, based on the access request parameters.

Sixth, sudo can send logs to syslog for some sense of consolidated log management, but the problem is that you may still need to parse the sudo entries from other important logs, making it much more complex to produce required audit and compliance reports.
If the sudoers file is centralized using an LDAP server, and policies are enforced based on user names, organizations can achieve some level of security and consistency. However, when policies mix the security considerations (i.e., using usernames and group names), an organization can end up with conflicting access policies, resulting in growing maintenance issues and a weakened security environment.

There are other options to sudo for controlling privileged account access. A good example is the solution from Fox Technologies, FoxT ServerControl, part of an Enterprise Access Management suite of solutions. FoxT ServerControl transparently grants privileged access permissions, without sharing privileged accounts passwords, based on fine-grained access rules that adapt based on the factors of the access request: who is asking, from where, to where, using what protocol, when, what they want to do.

FoxT ServerControl goes well beyond the open source sudo utility to further optimize access management including enabling a holistic blending of authentication with authorization, offering centralized management of SSH host keys, and centralized management of passwords. All user access activity, including keystroke logs for sensitive sessions, is also automatically consolidated. Audit friendly reports are readily created from that consolidated data using FoxT Report Manager. Pre-built report sets for SOX and other key regulations such as PCI, HIPPA, and FER/NERC are also available to further reduce the cost of compliance.

By centrally maintaining access across diverse Unix/Linux physical and virtual systems, as well as Windows-based servers, and automatically consolidating user activity logs and producing compliance-ready reports, organizations using FoxT are reducing the risk of insider fraud, achieving regulatory compliance with less effort, as well as optimizing security & system administrator productivity.

Following is a comparison between Sudo and FoxT ServerControl's privileged access management capabilities:

OpenWare/Sudo	FoxT ServerControl
Access is allowed if the local sudoers file permits it, creating a security risk	There is no access to privileged accounts unless a fine-grained access rule is defined and then granted in real-time by FoxT ServerControl.
Config files must be manually copied to each local machine (or the files must be transfer periodically) for any change. Can be viewed as "static trust" by auditors, which is undesirable. Using sudo, an admin can create a method for each host to get the file it needs with "least privilege", but it isn't automatic, and it isn't easy to do right.	Access policies reside in a central database and updates are instantaneous. Enables organizations to pass audits since the local trust relationships do *not* reside in operating system files that could then be exploited and create a security risk.
The default with sudo is to put user activity logs on the local server. You could configure sudo to send logs to remote server, but that requires configuration effort and it is more complex.	Audit log are automatically stored on the master security server in a directory only readable by root. Most importantly, audit logs are stored in a way that local administrators cannot modify them.
Logs only invoking command line.	Full keystroke logging of input and resulting output available.
Statically defined permissions.	If and when an access policy is changed by security officers, it is then available to all FoxT protected servers immediately. As well, authorization/ access permissions adapt based on time of day, day of week, on which tty it is run, who is making request, from where, etc.
Cumbersome to implement, manage, and maintain.	The FoxT system is managed centrally and changes are immediately implemented throughout the domain. System admin doesn't spend hours building and distributing sudoers files. Simple method of creating and managing access routes. Regular updates to new functionality under maintenance programs.
Control of sudoers config file is not limited by privilege. This is a problem where a server is running critical apps, and sudo controls what privileged commands can be run against privileged data. If the control is the sudoers file, then the control issue are: • Who has the right to update the file? • Logging of file update • Link to change management processes • Versioning of file • Exclusion by authority and role to RW file • Access to RW the file on the server by some other priv. route (i.e. su)	FoxT centrally controls the authorization to utilize privileged accounts, without sharing the privileged account password, based on fine-grained access rules. In addition, administration of these authorizations is separately controlled with granular sub-administration.

Summary: While sudo provides an adequate method for privileged access management in small server estates, it is a cumbersome utility with the potential to create increased exposure to insider fraud for organizations trying to control access across large, diverse server infrastructures. In addition to requiring highly paid system administrators to spend a great deal of time building, and distributing sudoers files, sudo also forces you to rely on the individual expertise of your system administrator to plan and implement sudo in such a way that provides "least privileges". The new privileged access management solutions available on the market today provide highly efficient and effective alternatives to sudo. Using these modern approaches, you will be able to reduce the risk of insider fraud, streamline regulatory compliance, and greatly reduce the effort required to administer your server estates.

FOXT.

www.foxt.com • 883 North Shoreline Blvd. Building D, Suite 210 Mountain View, CA 94043 USA • 650.687.6300