

# Privileged Remote Access for **Third-Party Users**

## Secure Vendor Access With Full Session Governance

### The Third-Party Access Risk Problem

Third-party vendors, contractors, and partners are essential to modern operations. However, granting third-party users remote access to critical applications and privileged systems without visibility into what those users are doing creates governance and compliance challenges like:

- ⊗ Vendors retaining over-privileged access far beyond what their role requires
- ⊗ A lack of approval workflows for sensitive access requests
- ⊗ Recorded and unauditible privileged session activities

When third-party data breaches occur, organizations can't quickly identify what was accessed, who accessed it, or what actions were taken.

### Why VPN & Traditional Access Fail for Privileged Vendors

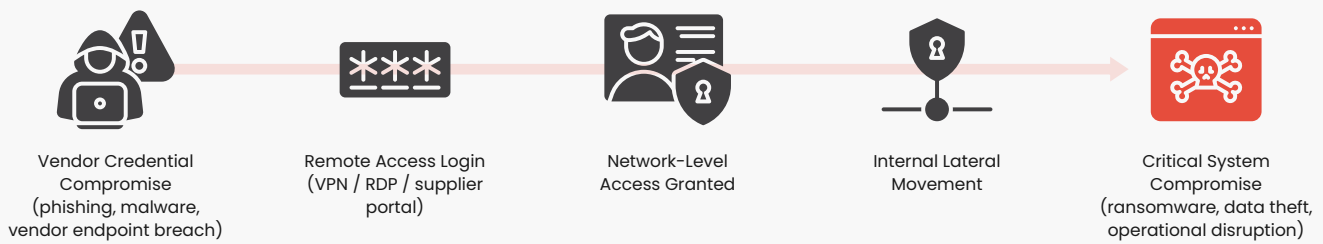
Unfortunately, VPN solutions were designed for internal networks, not vendor management. They grant users network-level access rather than controlled application-level access, so contractors get visibility into infrastructure they shouldn't see. If these credentials are stolen or leaked, attackers can log in via the VPN just like legitimate users and gain network-level access to internal systems.

Safous Privileged Remote Access eliminates network-level vendor access and replaces it with governed, identity-based application sessions. This approach enables organizations to control, monitor, and audit vendor and contractor access to sensitive systems – without requiring complex infrastructure such as VPNs, jump servers, or multi-module PAM deployments.

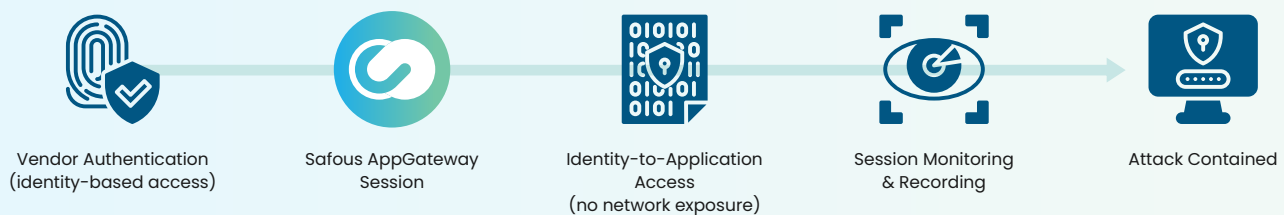
**The percentage of data breaches involving a third party doubled to 30% in 2025.**

<https://www.verizon.com/business/resources/T4d5/reports/2025-dbir-data-breach-investigations-report.pdf>

## Typical Attack Path Using VPN-Based Vendor Access



## Safous Privileged Remote Access



Attackers increasingly exploit trusted vendor credentials to conduct supply-chain attacks. Safous prevents this by eliminating network-level vendor access and enforcing governed application sessions.

## Safous Core Capabilities



### Granular Privileged Access Control:

Third-party users authenticate via identity provider integration and receive only the access they require. Access is limited to the application layer, not network-level, which eliminates lateral movement risk.



### Just-in-Time Access Provisioning:

Contractors and vendors don't require standing privileged accounts. Safous enforces time-bound, role-limited access provisioning with automated approval workflows, and access expires automatically.



### Full Session Visibility, Auditability, & Accountability:

Every privileged remote access session is recorded and auditable, with SSH/RDP activity, action-level commands, and session transcripts logged and retained entirely within your environment.



### Reduced Third-Party Risk Without VPN:

Safous provides controlled application-level access without requiring network flattening via VPN. This means no shared credentials, software on vendor devices, or network topology changes.

# Why Safous for Vendor Access?



**Agentless Vendor Access:** Safous enables secure vendor access without requiring agents, software installation, or device management on vendor laptops, reducing complexity and ensuring legitimate users can still quickly access your environment.



**Purpose-Built Solution:** Unlike traditional PAM tools that require multiple modules or additional infrastructure to support remote vendor access, Safous is purpose-built to deliver privileged remote access governance in a single streamlined platform.

## Secure Vendor Access With Full Session Governance

- Stronger access security
- Real-time visibility & control
- Centralized access control
- Improved user experience
- Support for compliance requirements
- Fast deployment



Safous enables access governance for third-party vendors and contractors in an agentless, unified platform designed specifically for secure privileged remote access.

GET STARTED TODAY. [➔](#)

