Módulo 2. Cifrado y acceso seguro

CÓRDOBA CREA FUTURO

1. Protocolo de red criptográfico Secure Shell (SSH) y secretos

Políticas de claves y agentes

Secure Shell, conocido como SSH, es un protocolo criptográfico diseñado para permitir acceso remoto seguro a sistemas a través de redes potencialmente inseguras. Su modelo de autenticación basado en criptografía de clave pública reemplaza el uso de contraseñas estáticas por un mecanismo más robusto y controlable. Gestionar adecuadamente claves SSH constituye una responsabilidad directa cuando se administra infraestructura crítica.

La autenticación por clave pública se basa en un par de claves: una privada, que permanece bajo control del usuario o del sistema cliente, y una pública, que se instala en el servidor dentro del archivo authorized_keys. Cuando se establece la conexión, el servidor verifica que el cliente posee la clave privada correspondiente a la clave pública registrada. Este modelo reduce el riesgo de ataques de fuerza bruta asociados a contraseñas débiles o reutilizadas (SSH.com, s. f.).

Sin embargo, la seguridad del esquema no depende únicamente de la fortaleza criptográfica del algoritmo utilizado. La gestión del ciclo de vida de las claves resulta determinante. Las buenas prácticas de administración de claves SSH incluyen mantener inventarios actualizados, establecer políticas de rotación periódica y eliminar claves asociadas a usuarios que ya no requieren acceso (SSH.com, s. f.). La proliferación descontrolada de claves públicas en servidores genera un riesgo difícil de auditar, especialmente en entornos con múltiples administradores o automatizaciones.

Otro aspecto relevante es el uso de agentes SSH. Un agente SSH permite almacenar claves privadas en memoria para evitar reingresarlas en cada conexión. Aunque mejora la usabilidad, también introduce riesgos si se utiliza sin restricciones adecuadas. El reenvío de agente, conocido como agent forwarding, puede permitir que una sesión comprometida utilice la clave almacenada para acceder a otros sistemas. Las prácticas recomendadas sugieren restringir el uso de agent forwarding únicamente a escenarios justificados (Hostinger, s. f.).

Deshabilitar la autenticación por contraseña en servidores donde se utiliza autenticación por clave pública constituye otra medida de control. Esta configuración reduce la

superficie de ataque frente a intentos automatizados de acceso. Mantener ambos métodos habilitados amplía innecesariamente las posibilidades de explotación.

La siguiente tabla resume controles asociados a una política de gestión de claves SSH.

Tabla 1: Política de gestión de claves SSH

Control	Riesgo mitigado	Evidencia esperada
Inventario centralizado de claves	Accesos no identificados	Registro documentado de claves autorizadas
Rotación periódica	Persistencia indebida de accesos	Política formal de rotación
Deshabilitación de autenticación por contraseña	Ataques de fuerza bruta	Configuración en sshd_config
Restricción de agent forwarding	Movimiento lateral no controlado	Política definida y configuración aplicada

Fuente: elaboración propia.

La ausencia de políticas formales conduce con frecuencia a la acumulación de claves antiguas que permanecen activas indefinidamente. Este escenario dificulta determinar quién tiene acceso efectivo a un sistema determinado. La administración responsable implica revisar periódicamente las claves autorizadas y vincular cada una con una identidad concreta.

Gestionar claves SSH no consiste únicamente en generar pares criptográficos. Requiere establecer procesos de control, documentación y auditoría que permitan sostener un modelo de acceso coherente con la arquitectura de seguridad definida para la organización.

Restricción por comandos/hosts

La autenticación mediante claves públicas no implica necesariamente acceso irrestricto al sistema. Una práctica técnica relevante consiste en limitar qué puede hacerse con una clave específica y desde qué origen puede utilizarse. El archivo authorized_keys permite definir opciones adicionales

asociadas a cada clave pública, incorporando controles que refuerzan el principio de mínimo privilegio.

Cuando se agrega una clave pública al archivo authorized_keys, es posible anteponer directivas que restringen su comportamiento. Entre las opciones disponibles se encuentran command="...", from="...", no-pty, no-port-forwarding y no-agent-forwarding. Estas directivas permiten limitar la ejecución a comandos específicos, restringir direcciones IP de origen o impedir el uso de túneles y sesiones interactivas (Red Hat, s. f.).

La opción command="..." obliga a que, independientemente de la solicitud del cliente, solo se ejecute el comando definido en la configuración. Este mecanismo resulta útil para automatizaciones controladas, como tareas de respaldo o despliegues, donde no se requiere acceso completo a la shell. De esta manera, incluso si la clave se utiliza desde un cliente comprometido, el alcance de la acción queda acotado.

La opción from="IP" o from="rango" permite restringir el uso de la clave a direcciones específicas. Este control agrega una capa adicional, limitando el acceso a determinados hosts, como bastion servers o redes internas previamente definidas. En entornos corporativos, esta práctica reduce la

probabilidad de que una clave comprometida sea utilizada desde ubicaciones no autorizadas.

Deshabilitar la asignación de pseudo-terminal mediante no-pty impide que el usuario obtenga una sesión interactiva. Esta restricción resulta apropiada cuando la clave está destinada exclusivamente a procesos automatizados. De forma similar, no-port-forwarding y no-agent-forwarding impiden la creación de túneles o el reenvío de agentes, reduciendo la posibilidad de movimientos laterales dentro de la red.

El siguiente cuadro resume opciones frecuentes y su aplicación práctica.

Tabla 2: Opciones de restricción en authorized_keys

Opción	Función	Escenario de uso
command="..."	Limita la ejecución a un comando específico	Automatización controlada
from="IP/rango"	Restringe el origen de conexión	Acceso desde bastion host

no-pty	Impide shell interactiva	Tareas programadas
no-agent-forwarding	Evita reenvío de agente SSH	Entornos segmentados

Fuente: elaboración propia.

Un error habitual consiste en instalar claves públicas sin aplicar restricciones adicionales, otorgando acceso completo a la shell por defecto. Esta práctica contradice el principio de mínimo privilegio y amplía innecesariamente el alcance potencial de un compromiso.

Configurar restricciones en authorized_keys no reemplaza otros controles, pero complementa la política general de acceso. Integrar estas opciones dentro de una estrategia de administración coherente contribuye a reducir riesgos asociados a accesos remotos y automatizaciones.

Sistema de administración centralizada de secretos Vault

En entornos donde múltiples servicios requieren credenciales, tokens o claves criptográficas, almacenar secretos de forma distribuida en archivos de configuración o variables de entorno incrementa el riesgo de exposición. La administración centralizada de secretos surge como respuesta a este problema. HashiCorp Vault es una herramienta diseñada para almacenar, proteger y distribuir secretos de manera controlada.

Vault funciona como una bóveda cifrada que almacena información sensible, incluyendo contraseñas, claves privadas, certificados y credenciales dinámicas. Su arquitectura se basa en un almacenamiento cifrado donde los datos se protegen mediante un mecanismo de sellado. Cuando Vault inicia, permanece en estado sellado hasta que se proporcionan un número determinado de claves de desbloqueo. Este proceso se conoce como unseal y requiere la participación de múltiples partes cuando se configura bajo el esquema de control compartido (HashiCorp, s. f.).

El modelo de sellado y desellado puede generar dificultades conceptuales. El estado sellado implica que el almacenamiento cifrado no es accesible. Para desbloquearlo, se requiere reconstruir la clave maestra a partir de fragmentos distribuidos entre responsables designados. Este

enfoque reduce el riesgo de acceso unilateral y se alinea con principios de separación de funciones.

Vault también implementa políticas de acceso basadas en tokens. Cada entidad autenticada recibe un token con permisos específicos definidos por políticas declarativas. De este modo, el acceso a secretos puede limitarse a determinadas rutas o tipos de operaciones. Esta granularidad permite aplicar el principio de mínimo privilegio en la gestión de credenciales.

Una característica relevante es la generación dinámica de secretos. En lugar de almacenar credenciales estáticas para bases de datos o servicios, Vault puede generar credenciales temporales con tiempo de vida limitado. Una vez vencidas, estas credenciales dejan de ser válidas automáticamente. Este modelo reduce la exposición asociada a secretos persistentes y simplifica procesos de revocación.

Centralizar secretos no elimina la necesidad de proteger los sistemas que acceden a la bóveda. Si un servidor comprometido posee permisos amplios en Vault, puede extraer información sensible. Por ello, la definición adecuada de políticas y la segmentación de accesos constituyen aspectos clave en la implementación.

Adoptar una solución de administración centralizada implica también registrar accesos y operaciones realizadas sobre los secretos. Vault permite auditar solicitudes y generar registros que facilitan la supervisión y el análisis posterior.

Gestionar secretos de manera distribuida sin control central dificulta la rotación y aumenta la probabilidad de exposición accidental. Incorporar una bóveda con políticas definidas, mecanismos de sellado y generación dinámica de credenciales contribuye a estructurar un modelo coherente de protección de información sensible.

Key Escrow: Almacenamiento de claves de cifrado de forma segura para recuperación

El uso de cifrado protege información frente a accesos no autorizados, pero también introduce el riesgo de pérdida de datos si las claves se extravían. El concepto de key escrow se refiere al almacenamiento controlado de claves criptográficas con el propósito de permitir su recuperación

en circunstancias específicas. La recomendación del NIST sobre gestión de claves aborda esta práctica dentro del marco de recuperación de claves y continuidad operativa (NIST, 2020).

Key escrow no equivale a realizar copias indiscriminadas de claves privadas. Implica establecer mecanismos formales de custodia que equilibren disponibilidad y protección. Si una clave se pierde sin posibilidad de recuperación, los datos cifrados pueden volverse inaccesibles de manera permanente. Sin embargo, almacenar claves sin controles adecuados puede facilitar accesos indebidos.

El NIST SP 800-57 Parte 1 describe que la recuperación de claves debe implementarse únicamente cuando exista una necesidad organizacional justificada y bajo controles estrictos de acceso (NIST, 2020). Entre estos controles se encuentran la separación de funciones, el uso de múltiples custodios y el registro detallado de cualquier operación de recuperación.

En entornos corporativos, los escenarios que justifican un esquema de recuperación incluyen la pérdida de credenciales por parte de un usuario, la salida de un empleado con acceso a información cifrada o la necesidad de acceder a datos en el marco de una investigación interna.

En estos casos, disponer de un mecanismo formal permite evitar la pérdida irreversible de información.

La implementación práctica puede basarse en dividir la clave de recuperación en múltiples fragmentos distribuidos entre responsables designados. Este modelo, similar al utilizado en el proceso de unseal de Vault, reduce el riesgo de acceso unilateral. Solo cuando se reúnen los fragmentos requeridos puede reconstruirse la clave necesaria para acceder a la información cifrada.

La siguiente tabla resume escenarios comunes asociados a la recuperación de claves y los controles recomendados.

Tabla 3: Escenarios de recuperación de claves y controles asociados

Escenario	Riesgo si no existe recuperación	Control recomendado
Pérdida de clave por usuario	Pérdida irreversible de datos cifrados	Procedimiento formal de recuperación
Salida de empleado	Acceso persistente o datos inaccesibles	Revocación + custodia controlada

Investigación interna	Imposibilidad de acceder a información relevante	Control dual y registro de operaciones

Fuente: elaboración propia.

Implementar key escrow exige definir políticas claras sobre quién puede autorizar una recuperación, bajo qué condiciones y cómo se documenta el proceso. La ausencia de controles formales puede transformar un mecanismo de continuidad en un punto de vulnerabilidad.

La gestión responsable del cifrado no consiste únicamente en proteger claves frente a terceros, sino también en prever escenarios de pérdida y establecer mecanismos de recuperación alineados con estándares de gestión de claves.

CONTINUAR

2. VPN e IPsec

Creación de redes privadas virtuales (VPN) con WireGuard: claves y peers

Una red privada virtual, conocida como VPN, permite establecer un túnel cifrado a través de una red pública, encapsulando tráfico para proteger su confidencialidad e integridad. Cuando diseñas arquitectura de acceso seguro, la VPN se convierte en un componente central para conectar usuarios remotos, sucursales o servicios distribuidos.

WireGuard es un protocolo moderno de VPN que se caracteriza por su diseño minimalista y su implementación eficiente. A diferencia de soluciones tradicionales que incorporan múltiples modos de configuración, WireGuard adopta un modelo simplificado basado en criptografía moderna y un esquema de pares de claves públicas y privadas para identificar nodos (WireGuard, s. f.).

Cada participante en una red WireGuard, denominado peer, posee un par de claves. La clave privada permanece en el

nodo local, mientras que la clave pública se comparte con los demás peers autorizados. La autenticación se basa en el conocimiento previo de estas claves públicas. No existe negociación compleja de certificados; el modelo se fundamenta en la asociación explícita entre claves y direcciones permitidas.

El concepto de peers implica que cada nodo conoce explícitamente con quién puede comunicarse. En la configuración se definen las claves públicas de los peers remotos y las redes internas que pueden alcanzarse a través del túnel. Este diseño reduce la superficie de configuración y facilita el análisis del tráfico permitido.

El establecimiento del túnel no solo cifra la comunicación, sino que también define rutas específicas. WireGuard utiliza tablas de enrutamiento del sistema operativo para dirigir el tráfico destinado a determinadas subredes hacia la interfaz virtual creada por el protocolo. Comprender esta lógica de enrutamiento resulta necesario para evitar configuraciones incorrectas que interrumpan conectividad o generen bucles.

La simplicidad del protocolo también impacta en el rendimiento. WireGuard utiliza algoritmos criptográficos modernos con bajo costo computacional, lo que favorece su adopción en dispositivos con recursos limitados. Sin

embargo, la simplicidad no elimina la necesidad de una correcta gestión de claves. Si una clave privada se compromete, el peer asociado debe ser revocado y reemplazado mediante la generación de un nuevo par de claves.

Implementar WireGuard requiere definir claramente qué redes internas serán accesibles a través del túnel y qué nodos estarán autorizados. La ausencia de una planificación de direcciones puede generar superposición de subredes y problemas de conectividad.

Diseñar túneles cifrados con WireGuard implica integrar gestión de claves, definición de peers y configuración de enrutamiento en un esquema coherente. La arquitectura resultante debe permitir acceso controlado sin exponer servicios innecesarios a redes públicas.

OpenVPN: perfiles y MTU

OpenVPN es una solución de VPN ampliamente utilizada que opera bajo un modelo cliente-servidor. A diferencia de

WireGuard, que adopta un enfoque minimalista, OpenVPN ofrece una amplia variedad de opciones de configuración, incluyendo distintos modos de autenticación y encapsulación. Esta flexibilidad permite adaptarse a múltiples escenarios, pero también exige comprender con precisión los parámetros involucrados.

La configuración de OpenVPN suele distribuirse mediante archivos de perfil que contienen directivas relativas a certificados, claves, direcciones del servidor, puertos y parámetros de red. Estos perfiles determinan cómo el cliente establece el túnel cifrado y cómo se enruta el tráfico a través de la interfaz virtual creada durante la conexión.

Uno de los aspectos técnicos que suele generar dificultades es el ajuste del MTU, Maximum Transmission Unit. El MTU define el tamaño máximo de paquete que puede transmitirse sin fragmentación. Cuando se encapsula tráfico dentro de un túnel VPN, se agrega información adicional al paquete original. Si el tamaño resultante excede el MTU permitido por la red subyacente, se produce fragmentación o pérdida de paquetes.

La documentación comunitaria de OpenVPN explica que una configuración inadecuada del MTU puede generar problemas de rendimiento, desconexiones intermitentes o

imposibilidad de transmitir ciertos tipos de tráfico (OpenVPN Community, s. f.). Un MTU demasiado alto puede provocar fragmentación excesiva, mientras que uno demasiado bajo reduce la eficiencia del canal.

Comprender el impacto del MTU requiere considerar que cada capa adicional de encapsulación incrementa el tamaño total del paquete. Ajustar correctamente este parámetro implica analizar la red subyacente y realizar pruebas para identificar el valor óptimo que evite fragmentación innecesaria.

El siguiente cuadro resume posibles efectos asociados a configuraciones de MTU.

Tabla 4: Impacto del MTU en rendimiento de VPN

Configuración	Efecto	Riesgo asociado
MTU demasiado alto	Fragmentación de paquetes	Pérdida de rendimiento y estabilidad
MTU demasiado bajo	Subutilización del ancho de banda	Incremento de latencia

MTU ajustado correctamente	Flujo estable de tráfico	Optimización del rendimiento

Fuente: elaboración propia.

Además del MTU, OpenVPN puede configurarse con distintos métodos de autenticación, incluyendo certificados digitales y credenciales adicionales. La correcta distribución y protección de estos perfiles resulta necesaria para evitar exposición de claves privadas o datos sensibles.

Diseñar una implementación robusta implica validar la conectividad bajo distintas condiciones de red y monitorear el comportamiento del túnel. Los problemas asociados a fragmentación pueden no ser evidentes en pruebas iniciales, pero manifestarse bajo cargas específicas.

El ajuste cuidadoso de perfiles y parámetros de red contribuye a mantener estabilidad y rendimiento. Integrar pruebas sistemáticas dentro del proceso de

despliegue permite anticipar fallas antes de que afecten a usuarios finales.

Conjunto de protocolos de Seguridad de Internet IPsec e implementación StrongSwan: escenarios

IPsec es un conjunto de protocolos diseñados para asegurar comunicaciones a nivel de red. A diferencia de soluciones que operan en capas superiores, IPsec protege paquetes IP directamente, proporcionando confidencialidad, integridad y autenticación. Su aplicación resulta frecuente en conexiones sitio a sitio y en entornos donde se requiere integración con infraestructura de red corporativa.

El conjunto IPsec se compone principalmente de dos protocolos: Authentication Header (AH) y Encapsulating Security Payload (ESP). AH proporciona autenticación e integridad, pero no cifrado. ESP, en cambio, puede ofrecer cifrado además de integridad y autenticación. En implementaciones actuales, ESP es el componente más utilizado debido a la necesidad de confidencialidad en comunicaciones remotas (StrongSwan, s. f.).

La negociación de parámetros criptográficos en IPsec se realiza mediante el protocolo IKE, Internet Key Exchange. IKE establece asociaciones de seguridad entre pares y define algoritmos de cifrado, autenticación e intercambio de claves. El diseño modular de IPsec permite seleccionar distintos algoritmos conforme a políticas definidas en cada extremo.

IPsec puede operar en modo transporte o en modo túnel. En modo transporte, solo se protege la carga útil del paquete IP, manteniendo intacta la cabecera original. Este modo suele utilizarse en comunicaciones host a host. En modo túnel, se encapsula el paquete IP completo dentro de un nuevo paquete, protegiendo tanto la carga como la cabecera original. Este modo es habitual en conexiones sitio a sitio entre redes corporativas.

StrongSwan es una implementación ampliamente utilizada de IPsec que permite configurar túneles mediante archivos declarativos. Su documentación describe la configuración de asociaciones de seguridad, políticas criptográficas y autenticación mediante certificados o claves precompartidas (StrongSwan, s. f.). Integrar certificados digitales dentro de IPsec permite mantener coherencia con la infraestructura PKI previamente definida.

El siguiente cuadro resume diferencias entre modos de operación.

Tabla 5: Comparación IPsec modo transporte vs modo túnel

Modo	Uso típico	Nivel de encapsulación
Transporte	Comunicación host a host	Protege únicamente la carga útil
Túnel	Conexión sitio a sitio	Encapsula el paquete IP completo

Fuente: elaboración propia.

Implementar IPsec en entornos corporativos requiere coordinar configuraciones en ambos extremos del túnel. La discrepancia en parámetros de cifrado o autenticación impide el establecimiento de la asociación de seguridad. Por ello, documentar políticas criptográficas y mantener consistencia entre dispositivos resulta necesario.

El uso de certificados en lugar de claves precompartidas mejora la escalabilidad en infraestructuras con múltiples nodos. Las claves precompartidas pueden resultar adecuadas en escenarios limitados, pero presentan desafíos de gestión cuando la cantidad de pares aumenta.

Diseñar túneles IPsec implica evaluar topología de red, requisitos de rendimiento y compatibilidad con equipos existentes. Integrar IPsec con mecanismos de autenticación robustos contribuye a establecer canales seguros entre redes sin exponer tráfico sensible a interceptación.

Postura de seguridad del cliente

Establecer un túnel cifrado no garantiza por sí mismo un entorno seguro. Si el dispositivo que se conecta a la red corporativa se encuentra comprometido, el canal protegido puede convertirse en un medio para introducir amenazas. Por ello, la evaluación de la postura de seguridad del cliente constituye un componente relevante dentro del diseño de acceso remoto.

La guía del NIST sobre VPN basadas en SSL destaca la necesidad de considerar controles del lado del cliente antes de permitir el acceso a recursos internos (NIST, 2008). Estos controles pueden incluir verificación de software antivirus activo, estado de actualización del sistema operativo y configuración adecuada del firewall local. Permitir conexiones sin evaluar estas condiciones incrementa la probabilidad de que dispositivos vulnerables accedan a la red interna.

El concepto de posture assessment implica evaluar automáticamente ciertos parámetros del dispositivo antes de conceder acceso completo. En entornos empresariales, esta evaluación puede integrarse con soluciones de gestión de endpoints que verifican cumplimiento de políticas. Si un dispositivo no cumple requisitos mínimos, puede limitarse su acceso o redirigirse a una red segmentada.

Uno de los errores frecuentes consiste en implementar VPN sin ningún tipo de validación del cliente. Esta práctica parte del supuesto de que el cifrado del túnel es suficiente para garantizar seguridad. Sin embargo, un equipo sin parches de seguridad o sin protección contra malware puede representar un riesgo mayor que la propia transmisión de datos en redes públicas.

La siguiente tabla resume controles asociados a la postura de seguridad del cliente.

Tabla 6: Controles de postura de seguridad del cliente

Control	Objetivo	Riesgo mitigado
Antivirus activo y actualizado	Detectar software malicioso	Infección de la red interna
Sistema operativo actualizado	Corregir vulnerabilidades conocidas	Explotación remota
Firewall habilitado	Controlar tráfico entrante y saliente	Acceso no autorizado
Autenticación fuerte	Verificar identidad del usuario	Verificar identidad del usuario

Fuente: elaboración propia.

Implementar controles de postura no implica desconfianza hacia los usuarios, sino reconocimiento de que los dispositivos pueden verse afectados por vulnerabilidades. Integrar estas verificaciones dentro de la arquitectura de acceso seguro contribuye a reducir la probabilidad de incidentes derivados de equipos comprometidos.

La evaluación de la postura debe complementarse con políticas claras sobre actualización y protección de dispositivos. Un canal cifrado protege la comunicación, pero la seguridad global depende también del estado del punto final que participa en ella.

CONTINUAR

Cierre inegrador

Diseñar acceso seguro implica integrar múltiples componentes en una arquitectura coherente. SSH protege el acceso administrativo a sistemas individuales; la administración centralizada de secretos organiza la custodia de credenciales; las VPN establecen túneles cifrados que conectan redes y usuarios remotos; IPsec refuerza comunicaciones a nivel de red; y la evaluación de postura del cliente agrega un control previo al acceso.

Cada uno de estos mecanismos responde a una dimensión distinta del problema de acceso. SSH gestiona autenticación y control de comandos en servidores. Vault estructura la protección y distribución de secretos. WireGuard y OpenVPN crean canales cifrados sobre redes públicas. IPsec encapsula tráfico entre redes completas. La verificación de postura introduce una evaluación del estado del dispositivo antes de permitir su integración al entorno corporativo.

La coherencia arquitectónica surge cuando estas herramientas no se implementan de manera aislada, sino articuladas bajo una política común. Por ejemplo, utilizar certificados emitidos por una PKI corporativa tanto para SSH como para IPsec refuerza consistencia en la gestión de identidad. Centralizar secretos evita que claves privadas se distribuyan de forma descontrolada. Restringir comandos en SSH reduce el impacto potencial de una clave comprometida.

La responsabilidad del arquitecto de acceso seguro consiste en equilibrar protección, disponibilidad y administración eficiente. Un túnel cifrado mal configurado puede afectar rendimiento; una clave sin control puede otorgar acceso excesivo; una VPN sin evaluación de postura puede permitir ingreso de dispositivos inseguros.

Construir acceso seguro no se limita a aplicar configuraciones técnicas. Implica definir políticas, establecer controles verificables y documentar decisiones. Integrar cifrado, autenticación y monitoreo en un esquema coherente permite reducir riesgos asociados a fugas de información y accesos no autorizados.

Do & Don’ts en SSH y acceso remoto

La gestión de acceso remoto requiere traducir principios técnicos en prácticas operativas concretas. El análisis de buenas prácticas en SSH destaca una serie de acciones que reducen riesgos asociados a autenticación, exposición de claves y movimientos laterales dentro de la red (Teleport, s. f.). A continuación se presenta una síntesis estructurada de conductas recomendadas y prácticas que conviene evitar.

Do	Don't	Justificación técnica
Utilizar autenticación basada en claves públicas	Permitir autenticación por contraseña en servidores críticos	Reduce exposición a ataques de fuerza bruta
Restringir comandos en authorized_keys	Otorgar acceso completo a la shell por defecto	Aplica principio de mínimo privilegio
Rotar claves periódicamente y mantener inventario	Mantener claves activas indefinidamente sin control	Reduce persistencia de accesos no necesarios

Deshabilitar agent forwarding cuando no es requerido	Mantener claves activas indefinidamente sin control	Minimiza riesgo de movimiento lateral
Registrar y auditar accesos SSH	No conservar registros de conexión	Mejora trazabilidad y respuesta ante incidentes
Centralizar gestión de secretos	Distribuir claves manualmente en múltiples servidores	Reduce riesgo de fuga y descontrol de credenciales

Implementar estas prácticas implica definir políticas formales y procedimientos de revisión periódica. La autenticación por clave pública reduce riesgos frente a contraseñas débiles, pero si las claves se almacenan sin control o no se revocan oportunamente, el riesgo se traslada a otro punto.

Restringir comandos mediante directivas en authorized_keys limita el alcance operativo de cada clave. Este control resulta especialmente relevante cuando se utilizan claves para automatizaciones. Permitir acceso completo sin restricciones amplía innecesariamente el impacto potencial de un compromiso.

La rotación periódica de claves y el mantenimiento de inventarios actualizados permiten identificar accesos obsoletos o innecesarios. La ausencia de inventario dificulta determinar quién posee acceso efectivo a un servidor determinado.

El registro de conexiones y la revisión periódica de logs facilitan la detección de comportamientos anómalos. Sin evidencia documentada, la reconstrucción de eventos en caso de incidente se vuelve compleja.

Centralizar la gestión de secretos mediante herramientas especializadas contribuye a evitar la dispersión de claves privadas en múltiples sistemas. La distribución manual sin control formal incrementa la probabilidad de exposición accidental.

Adoptar estas prácticas no elimina todos los riesgos asociados al acceso remoto, pero establece un marco coherente de control que reduce superficies de ataque y mejora la capacidad de respuesta ante incidentes.

CONTINUAR

Referencias

HashiCorp. (s. f.). *Getting started with Vault*. https://developer.hashicorp.com/vault/tutorials/getting-started/getting-started-install

Hostinger. (s. f.). *Cómo configurar claves SSH*. https://www.hostinger.com/ar/tutoriales/como-configurar-claves-ssh

NIST. (2008). *Guide to SSL VPNs (SP 800-113)*. https://csrc.nist.gov/pubs/sp/800/113/final

NIST. (2020). *Recommendation for key management: Part 1 – General (SP 800-57 Rev. 5)*. https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final

OpenVPN Community. (s. f.). *MTU and fragments*. https://community.openvpn.net/MTU%20and%20Fragments

Red Hat. (s. f.). *Key-based authentication in SSH*. https://www.redhat.com/es/blog/key-based-authentication-ssh

SSH.com. (s. f.). *SSH key management best practices*. https://www.ssh.com/academy/iam/ssh-key-management

StrongSwan. (s. f.). *Introduction to IPsec*. https://docs.strongswan.org/docs/latest/howtos/ipsecProtocol.html

Teleport. (s. f.). *5 SSH best practices*. https://goteleport.com/blog/5-ssh-best-practices/

WireGuard. (s. f.). *Installation & setup guide*. https://www.wireguard.com/install/

CONTINUAR