A Secure Shell White Paper

What Financial Institutions Need to Know About Secure Shell

ssh®

TABLE OF CONTENTS:

About SSH Communications Security:

In 1995, the company's founder, Tatu Ylönen, invented the Secure Shell protocol, which soon became the gold standard for data-in-transit security. Today, Secure Shell is one of the most widely used protocols in the world and SSH Communications Security has grown to serve over 3,000 customers around the globe, including 7 of the Fortune 10.

Throughout our history, we have developed leading edge security solutions that enable organizations to protect against a rapidly growing threat landscape that includes both internal and external actors. Our platform based approach to Secure Shell deployment and management provides the only solution on the market that addresses the need for security, compliance and operational efficiency in today's complex enterprise environments.

Introduction

Secure Shell is a protocol and software suite used for securely transmitting data, application tunneling and remote systems administration. It comes preinstalled on Unix, Linux, IBM Mainframes and is available for Windows. It is deployed on millions of servers and is used in approximately 90% of enterprise IT environments. Systems administrators and application developers use Secure Shell for interactive access and it is possibly even more widely used for automated processes such as backups, monitoring applications, data transfers and systems management.

Secure Shell is also ubiquitous within the financial services industry. Banks, insurance companies, brokerages, credit unions, etc. all use Secure Shell for business processes critical to day-to-day operations and for bringing new services online. However, Secure Shell is often viewed as "part of the plumbing" and often does not get much attention from Compliance, Audit or Security. This mindset is rapidly changing. Poor controls over Secure Shell environments have contributed to costly data breaches and compliance violations. This white paper provides basic information on security and compliance issues as well as best practices for management and controls over Secure Shell environments within the financial services industry.

Secure Shell Authentication and Identity Management

At its core, Secure Shell is used for remotely logging into application and service accounts on remote servers. Secure Shell supports authentication methods such as passwords, tokens, digital certificates and public key. With public key authentication, a public key is configured on a server as an authorized key and the private key is stored on a client machine (which in itself is often a server computer) in a small file as an identity key. Private key files can be encrypted using a passphrase, but keys used for automation typically do not have a passphrase as the passphrase itself would need to be stored in a file or hardcoded in a script. Financial institutions have been using Secure Shell across their IT infrastructure for years. A large server infrastructure is likely to have hundreds of thousands (even millions) of public keys enabling access to system, user and application accounts.
If improperly managed, Secure Shell keys can be used by attackers to penetrate the IT infrastructure. The compromise of one private key can be leveraged to configure hard-to-notice backdoors, to bypass privileged access control solutions and to perpetrate large scale attacks and data breaches.
Lack of central management and control of Secure Shell deployments exposes financial institutions to three negative impacts:

1. Heightened risk of data breach and attacks on availability.
2. Violation of security mandates including PCI-DSS, SOX, BASEL III and Monetary Authority of Singapore Technology Risk Management Guidelines.
3. Excessive overhead costs resulting from manual processes used for Secure Shell deployment.

The fundamental source of these impacts is the decentralized manner in which Secure Shell is deployed. Lacking central controls, many financial institutions are unable to enforce basic identity and access management policies, such as ensuring the correct levels of authorization are assigned, credential management and protection, detection of policy violations and timely onboarding/offboarding.

Security Risks

The high level risks are summarized in table 1 below.

Contractors and employees who left years ago still have access to critical systems. This exposes the enterprise to data loss or other malicious activity.
Unneeded keys remain authorized on system, application and user accounts. Each public key based authorization creates an exposure in the event that the corresponding private key is compromised.
Unauthorized copies of private keys in circulation. In general, the holder of a private key can make copies, even if this violates company policy. Central monitoring and controls can reduce or even eliminate the potential for unauthorized copies being used.
Private keys not passphrase protected. Lack of policy enforcement over private key protection increases the risk of credentials being compromised.
Keys not rotated regularly or at all. Key rotation is a basic requirement for protecting credentials, just as most organizations require end users to regularly change their passwords.
Unintended escalation of access. Secure Shell configuration controls are needed to prevent users from escalating privileges or gaining access to other accounts.
Breakdown of separation of duties. This is a common issue in financial services – often caused by lack of controls over key authorizations that get propagated from development to production servers.
Unintended access between test and production environments.
Lack of visibility of trust relationships.
Inability to meet audit requirements. This can extend to basics such as reporting on all trust relationships and activity logging.
Human errors in manual key setup and removal process. This can result in unintended access being granted, or failure to remove authorizations when required.
Misconfigured Secure Shell software. This can allow users to violate policies such as prohibitions on VPN access in or out of the network.
Number of individuals authorized to create permanent trust relationships, resulting in breakdown of access controls.

Table 1: High level risks and issues of unmanaged Secure Shell deployments

Compliance Violations

Compliance mandates including PCI-DSS, SOX, BASEL III and others have common requirements regarding access controls, identity management and audit. These requirements pertain to any method or technology used for access to protected data, including Secure Shell.

Table 2 below summarizes the Secure Shell guidance for compliance to these requirements.

Common Compliance Requirements	Secure Shell Guidance
Identify all networks, network devices, and system components within scope of the compliance audit. This includes connectivity between the environment in scope and other networks.	SSH is used to create connectivity within and external to the scoped environment. This connectivity must be identified.
Document and provide business justification for use of all services, protocols, and ports allowed.	Secure Shell is used to create connectivity within and external to the scoped environment. This connectivity must be identified.
Firewall and router configurations must restrict connections between untrusted networks and any system components.	Secure Shell is often used for contractor and employees access from untrusted networks. Secure Shell enabled access must have proper restrictions.
Configuration standards must be documented and enforced for all system components. The standards must address all known security vulnerabilities and are consistent with industry-accepted system hardening standards.	Standards and enforcement of Secure Shell server software must be enforced.
Security policies and operational procedures for protecting sensitive data must be documented, in use, and known to all affected parties.	This includes procedures for controlling automated access to data (typically enabled via Secure Shell).
Policies and procedures for creating, assigning and tracking identities, their authorizations and credentials must be documented and enforced.	Secure Shell User Key based identities are subject to this requirement.
Describe the process for tracking User Keys.	Any person or process in possession of a private Secure Shell user key has access to accounts with the corresponding public key. Tracking and controlling distribution of these keys is a basic security requirement.
Policies and procedures must be in place for changing and removing identity based authorizations when users change roles or leave the organization. This includes employees, contractors and identities assigned to automated processes that have been discontinued.	Secure Shell based access must be modified or removed under these circumstances. All public keys assigned to the user must be removed.
Credentials such as user passwords, pass phrases and key pairs must be updated on a regular basis as a preventive measure against compromise.	A policy should be in place and enforced for regular key rotation.

Common Compliance Requirements	SSH Guidance
The number of people authorized to create identities should be controlled and kept to a minimum.	Creation of new Secure Shell authorizations should be centralized and with the same administrative checks and balances used for managing other forms of access and credentials.
Use strong cryptography and security protocols (for example, SSL/TLS, IPSec, Secure Shell, etc.) to safeguard sensitive data during transmission over open, public networks.	Ensure only trusted keys and certificates are accepted. Ensure the protocol in use only supports secure versions or configurations. Ensure the encryption strength is sufficient.
Restrictions must be in place to prevent authorized users from using access for an unauthorized purpose.	Restrictions must be in place for Secure Shell enabled access.
Use secured technologies such as SSH, S-FTP, SSL, or IPSec VPN to protect insecure services such as NetBIOS, file-sharing, Telnet, FTP, etc.	Ensure Secure Shell is deployed in a secure manner.
Ensure that all system components and software are protected from known vulnerabilities by installing applicable vendor-supplied security patches. Install critical security patches within one month of release.	Ensure Secure Shell software is installed and updated according to vendor or open source recommendations. No insecure versions of Secure Shell software or the protocol should be in use.
Monitor and audit access to critical data and systems.	Privileged activities such as those conducted by systems and applications administrators must be monitored.
Remove development, test and/or custom application accounts, user IDs, and passwords before applications become active or are released.	Secure Shell authentication keys used by development and test must be removed before applications are moved into production environments.
Separate development/test environments from production environments, and enforce the separation with access controls.	Automated and user interactive Secure Shell access should not be permitted between development/test environments and production environment.
Ensure that security policies and operational procedures for restricting access to confidential data are documented, in use, and known to all affected parties	The policy must address procedures for restricting access using Secure Shell keys if applicable.
Assign all users a unique ID before allowing them to access system components or confidential data.	Ensure provisions in place to prevent use of shared Secure Shell keys.

Table 2: Secure Shell guidelines for security compliance

Auditors responsible for assuring PCI-DSS, SOX, BASEL III and other mandates must include Secure Shell within the scope of their investigations.

Cost Implications

The manual processes associated with creating Secure Shell authorizations in a large, dynamic IT environment create a "hidden tax" on operations and in many cases can impact productivity. On a per server basis, the direct cost of manual Secure Shell key management is approximately $190/year/server. This does not include additional costs due to errors in key set ups or the costs of manual Secure Shell software configuration and updates.

Table 3 summarizes the costs typically borne by an IT infrastructure of 20,000 servers.

Number of servers enabled with Secure Shell access	20,000
Number of key set ups per year	10,000
Average time per set up	0.25 hours
Average number of systems per set up	10
Number of key removal operations per system per year	2
Time required per key removal	0.5 hours
Number of other key operations per server per year	4
Time required per other operation	0.15 hours
Hourly cost of system or security administrator	$59/hour
Total annual cost	$3,853,000

Table 3: Cost of manual SSH key administration.

Best Practices

Secure Shell can be deployed in a manner that addresses the risk, compliance and cost problems many financial services companies face. Best practices include the following:

1. Standardize the key configuration across the environment.
2. Authorized key file should not allow end user write access.
3. Centralized key provisioning (no more "self service" provisioning). Key provisioning should be centralized and limited to a much smaller number of root level administrators.
4. Cipher configuration – allow only strong ciphers and specified key lengths.
5. Require password protection for private keys.
6. Require logging of Secure Shell activity.
7. Ensure Secure Shell server will not execute if authorized keys file and home directory are insecure.
8. Prevent privilege escalation by process spawning.
9. Segregate system accounts from person accounts.
10. Use controls to limit Secure Shell access to specific commands and source addresses.
11. Rotate keys.
12. Remove unneeded User Keys.
13. Document key usage.
14. Regular audits.

SSH Communications Security provides products and services that enable these best practices. Through effective Secure Shell central management, audit and control, financial institutions lower risks of data breach, address compliance violations and reduce costs.

Our Solutions

Universal SSH Key Manager™

Universal SSH Key Manager (UKM) provides full central lifecycle management for Secure Shell identities. While central identity management is essential for strong security and compliance, UKM provides much more:

- Discovery. UKM finds all the Secure Shell identities in your network. Gaining visibility of all Secure Shell identities and trust relationships is essential for security and compliance.
- Monitoring. UKM tracks login activity and provides central reporting. Administrators can use this information to identify and remove unneeded authorized keys and gain new visibility into Secure Shell activity.
- Remediation. In many financial institutions, Secure Shell logins violate security restrictions such as authorizations between development and production environments. UKM provides administrators with the information and tools to safely bring the network into compliance without interfering with ongoing, daily business.
- Continuous Monitoring and Management. UKM continuously monitors for policy violations and provides alerts and automatically corrects policy violations. UKM prevents rogue authorizations from being added and has the reporting capabilities to demonstrate regulatory compliance.
- Configuration Management. Ensure Secure Shell configurations meet policy goals.
- Automation. Save operational overhead by automating processes for adding and deleting key based Secure Shell authorizations.

Universal SSH Key Manager 4 Step Program for Security and Compliance

CryptoAuditor™

Systems administrators, application administrators, developers and maintainers are considered “privileged users” because they have access to system and service accounts. Those accounts typically have access to valuable information which if stolen would cause great, potentially crippling financial loss. Privileged user activity is often a blind spot in network level defenses – particularly when Secure Shell is used for login and command execution. Firewalls – even “deep packet inspection” firewalls only see Secure Shell as a port 22 session. They have no visibility as to what commands, file transfers or applications are being run inside the Secure Shell session. Likewise, intrusion prevention systems (IPS) cannot detect malware embedded within a Secure Shell session nor can data loss prevention systems (DLP) detect data leakage when it takes place within Secure Shell. In short, Secure Shell is an effective vector for data breaches perpetrated by authorized but malicious insiders or by hackers who gain access to Secure Shell within the network. CryptoAuditor is an inline solution that transparently monitors, audits and controls activity within Secure Shell channels. Core capabilities include:

- Full recording and indexing of SSH, SFTP and RDP sessions.
- Searchable database of activity – including graphical sessions. This facilitates regular reporting as well as expedited investigations and forensics.
- Prevents misuse of Secure Shell connections. For example, users cannot take advantage of the Secure Shell protocol to set up unauthorized VPNs in and out of the network.
- Strong access controls. CryptoAuditor can assign access rights (where users can go and what they can do) based on directory based membership.
- Transparent. Users do not have change work processes, no changes needed in authentication or directory systems.
- Strengthens layered defenses. DLP, IPS and SIEMs gain visibility into encrypted communications.
- Compliance. Reporting and monitoring meet the needs of multiple compliance mandates.

Tectia© SSH Family

Data in Transit protection. It sounds simple – make sure that all high value data such as asset transfers and transaction information are encrypted, end-to-end. Sometimes it is not that simple. Legacy applications may not be set up for encryption and there is the complexity of establishing encrypted connections with 3rd parties. Lastly, there is the need for strong authentication and identity controls. The Tectia SSH family of Secure Shell products is designed specifically to meet the challenges that underlie the simple need to protect data in transit. Our multiplatform family of Secure Shell client server solutions make it easy to secure legacy application transfers (including in and out of IBM mainframe environments) as well as for newly developed applications. Five of the top ten global banks rely on Tectia SSH to secure their data in transit.

Our Company: SSH Communications Security

The transition from an unsecured Secure Shell deployment to a secure and compliant deployment requires personnel with the expertise to design, plan and implement a solution. Many organizations have a skills gap in this area. As the inventor of the Secure Shell protocol, SSH Communications Security is the industry expert. We have helped many leading, global financial institutions deploy Secure Shell to meet their goals of security, compliance, productivity and cost control. We understand the challenges and bring the knowledge and experience to help financial institutions take control over their Secure Shell infrastructure.

Further Reading

See our white papers on SSH Risk and Compliance at www.ssh.com

Several recent press reports on the need for better Secure Shell management:

http://www.grc-daily.com/dsp_getFeaturesDetails.cfm?CID=4080

http://www.networkworld.com/article/2601030/security0/nist-issues-best-practices-on-how-to-best-use-secure-shell-software.html

http://www.bankinfosecurity.com/ssh-keys-managing-risks-a-7248

ssh®

www.ssh.com