WHITE PAPER

Smartly Manage Secure Shell Keys to Strengthen Overall Security

Sponsored by: SSH Communications

Christian A. Christiansen
June 2015

IDC OPINION: MANAGING SECURE SHELL DEPLOYMENTS REDUCES RISK OF IDENTITY MISUSE AND ENABLES COMPLIANCE

Identity and access management (IAM) is a critical component of an enterprise's overall security strategy. This is especially true for the financial and retail sectors. IAM implementations have largely focused on gaining control over the massive number of users and automated systems by both enhancing governance and automating provisioning capabilities. IAM deployments have helped IT security teams gain significant control of and visibility into a broad range of users — from rank-and-file employees to C-level executives. However, this governance framework has largely ignored privileged users (systems and applications administrators) and the large number of automated application-to-application processes that drive day-to-day IT operations and business processes. The lack of visibility into privileged user and automated systems authentication is a gaping hole in the overall security architecture for many financial and retail entities.

Many IT professionals use Secure Shell (SSH) as a tool to securely log in to remote servers and devices, execute remote commands, and transfer files. In many large enterprises SSH is also utilized by applications to automatically move massive amounts of data from point to point on internal networks and to external business partners such as payment card processors, datacenters, government entities, disaster recovery sites and many more. A version of SSH is shipped with every version of Linux, Unix, and Mac OS X as well as on mainframe operating environments. Additionally, SSH is easy to deploy in Windows systems. With strong encryption capabilities, widespread availability, and high adoption rates, SSH is one of those hidden jewels that IT professionals use all the time.

How SSH is actually utilized and the serious security risks associated with a general lack of governance have probably "flown under the radar" of those not in the IT trenches. With that said, nothing within the protocol itself or the software that drives it needs to be fixed.

Rather, given the rapidly evolving compliance environment, the growing threat landscape (including threats from malicious insiders and sophisticated attackers), the amount of SSH-enabled automated data transfers and system access, and the extraordinary level of privileged access that SSH users have, senior IT managers and financial compliance officers should take a closer look at their SSH identities.

For IT managers, the internal processes involved in the provisioning, rotation, and removal of SSH keys are often clumsy and decentralized and thus provide very little visibility into who has access to various devices within the network. From an internal and external compliance standpoint, the ability to restrict access on a need to do/need to know basis is almost nonexistent even in highly regulated enterprises such as banks, brokerages, and insurance firms. To make matters worse, the ability to lock down the environment so that only highly trusted "SSH provisioning" administrators can modify access to the environment — thus preventing backdoor access with keys — and continuously monitor for policy violations, rogue keys, and anomalous activity is currently lacking.

A typical large enterprise with 10,000+ SSH-enabled hosts that has been using SSH for more than 10 years is likely to see millions of automated machine-to- machine logins enabled by over 1 million installed SSH keys. It is inevitable that a large number of these keys are no longer necessary and may be in the possession of people who should no longer have access to the resources authorized by those keys. The risk associated with poor SSH key management increases exponentially if a key granting root access to a large part of the network becomes compromised.

A hidden tax on enterprises is tied to the manual nature of SSH key management and remediation. IT staff have to perform many tasks manually that could be automated with the right software. By eliminating the time required to perform manual work and reducing the errors associated with those, enterprises can increase efficiency and free up IT staff for different tasks.

IDC believes that organizations must become engaged in addressing SSH key management as a component of the overall IAM strategy. This paper is designed to provide IT managers, especially those in the financial and retail sectors, a basic background material on SSH key management and access control issues, outline the attendant IAM risks, and describe how those risks can be mitigated.

METHODOLOGY

IDC created this paper in May 2015. Its premises and opinions are based on a combination of research sources, including IDC primary research on remote access concentrating on SSH, historical and current research through IDC customer and vendor surveys, and information on the subject of security as reported in blogs, the press, and other online information sources. In addition, IDC participated in briefings held by SSH Communications Security in order to gain an in-depth understanding of the company's products and business proposition.

IN THIS WHITE PAPER

In this white paper IDC highlights the IT challenge around IAM controls associated with Secure Shell-based access. It provides background on the protocol, including how it is used, and explains the major security issues. The paper describes how SSH Communications Security offers an SSH key management product that brings governance to Secure Shell access controls, including SSH key discovery, deployment, rotation, removal, and continuous monitoring, and how the solution can help organizations to gain operational efficiency, minimize risk and meet compliance objectives while reducing the overhead of manual provisioning processes.

SECURE SHELL WITHIN THE ENTERPRISE

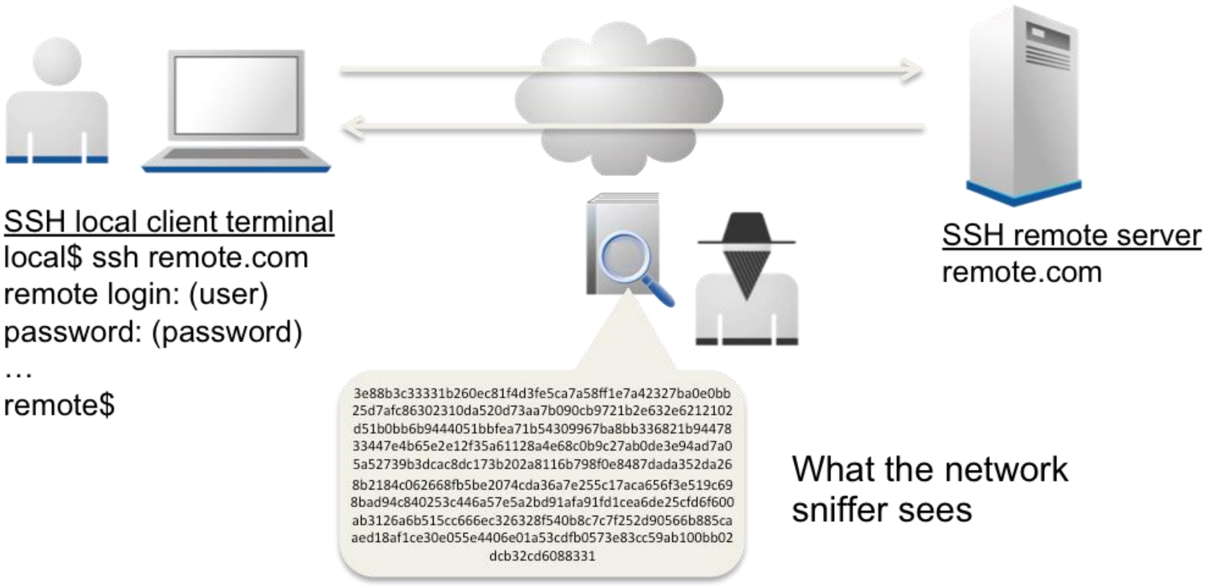
SSH has been in existence since 1995 when it was created to protect personal communications flow over an unsecured network. It became an Internet Engineering Task Force (IETF) standard in 1996. The original purpose of the protocol was to provide an encrypted tunnel that could secure remote log-ins, remote command executions, and file transfers. Those use cases are still widespread today. SSH has evolved into an automated and encrypted application-to-application data transfer workhorse, moving massive amounts of mission-critical and highly confidential information assets.

The SSH protocol uses public/private key pairs that are easily generated and stored. Access to the private key is generally accepted as proof for authentication. SSH works well for machine-to-machine communication because it enables authentication for scheduled and automated file transfers and other tasks where user interaction is not possible. Authentication is done automatically based on key associations established by a systems administrator.

Since its creation, SSH has become extremely popular with IT administrators, but it doesn't have much general exposure to those outside the IT trenches. Much of its popularity is driven by its pervasive availability and its ability to work independently without human intervention. A free version of SSH (OpenSSH) is available, and SSH is included in the Linux and Unix operating systems, contained within many Internet routers, and installed on a large share of Web servers. Much of its popularity is driven by its ease of use, and it is the "tool of choice" for systems and applications administrators.

FIGURE 1

Encrypted SSH Login Session to a Remote Host



```
local$ ssh remote.com
remote login: (user)
password: (password)
…
remote$
```

```
3e88b3c33331b260ec81f4d3fe5ca7a58ff1e7a42327ba0e0bb
25d7afc86302310da520d73aa7b090cb9721b2e632e6212102
d51b0bb6b9444051bbfea71b54309967ba8bb336821b94478
33447e4b65e2e12f35a61128a4e68c0b9c27ab0de3e94ad7a0
5a52739b3dcac8dc173b202a8116b798f0e8487dada352da26
8b2184c062668fb5be2074cda36a7e255c17aca656f3e519c69
8bad94c840253c446a57e5a2bd91afa91fd1cea6de25cfd6f600
ab3126a6b515cc666ec326328f540b8c7c7f252d90566b885ca
aed18af1ce30e055e4406e01a53cdfb0573e83cc59ab100bb02
dcb32cd6088331
```

Source: SSH Communications, 2015

SSH Keys: A Short Tutorial

SSH keys consist of a private and public key pair that is used to prove a user's identity during the SSH authentication process. The public key is also referred to as the "authorized key" and the private key is alternatively known as the "identity key."

SSH supports a self-service model for key deployment. SSH key pairs are created using a "key generation" program. Anyone, including automated services, can produce SSH keys. An SSH key pair is nothing more than a cryptographic key. It does not contain any creation information (date, originator, or owner) nor does it have a chain of trust, which is contained within a standard public key infrastructure certificate. The SSH key's public half is provided to all remote systems that the holder (user, device, or automated process) of the matching private key is allowed to connect with. For example, a systems administrator will publish his or her public key to all servers within his or her scope of responsibility. Private keys are controlled by and should be accessible to only the key owner. When a user, a device, or an automated process attempts to connect, the private key is used to verify the relationship between the published public key and the private key. However there is no verifiable link between the key and the identity of the user. All that is proven is that the user has possession of the private key. While authentication is based on the private key, the key itself is never transferred through the network during authentication. SSH private keys can be protected with a passphrase, but this type of protection is often not implemented or enforced for interactive SSH users (e.g., systems administrators) or for automated application-to-application processes.

Risks from Secure Shell Keys

The Secure Shell protocol is very well-suited to supporting automated processes and efficient systems administration. Authorization is initialized with the creation of a private and public key pair. The identity (private) key resides on the system requesting access to another system, and the corresponding authorized (public) key is installed on all the systems granting access. The key pairs that grant access to services are called SSH user keys. Key pairs can be created in a pure self-service model or can be created by authorized administrators when part of a corporate approval process. Regardless of deployment model, as organizations deploy SSH across their IT infrastructure, the process of key creation and distribution to target servers is repeated many times. Tracking is often haphazard, and basic controls, such as removal of key authorizations when an employee leaves the company, are lacking. The result is an extended, poorly managed access infrastructure built upon SSH user keys.

In addition to poor controls over public SSH user keys, there are few built-in security safeguards over SSH private keys. There is an option to include passphrase protection over private keys, but without central controls, end users are free to retain private keys without a passphrase. There are also no mechanisms to prevent users from making copies of private keys and using them after they leave the company. Private keys are also susceptible to theft by malware – enabling access to systems and data by external bad actors. It takes only one rogue, orphaned, or lost key to compromise the whole network.

An interesting unintended consequence of SSH is that an SSH connection can be used to bypass access control mechanisms such as password-based systems. If a system account (operating systems, middleware, databases, and applications for running processes) has a key association, a user can make a connection to the system account, circumventing the standard password-based authentication. This access is made possible because the SSH key association provides acceptable authentication.

In summary, identity and access management risks within Secure Shell implementations without an SSH key manager include:

- No visibility into purpose of key pairs
- Limited control over the creation of SSH keys
- Ease of copying and moving private keys
- Limited ability to identify and remove revoked, orphaned, and unauthorized keys
- Unused user keys that still grant access to critical hosts
- SSH key usage that circumvents IAM controls

Breach Examples

The value associated with SSH keys is high within the hacker community. Being able to steal SSH keys is a goldmine for attackers. As former teenage hacker and now security researcher Sean M. Bodmer said in a news article, "It's quite horrific what access you can get with an SSH key." He added that hackers could use abandoned keys to move through a secured computer network by hopping from server to server. "It's a domino effect" security breach, he said.

There have been studies to determine the actual level of interest hackers attach to gaining access to SSH keys. An independent blogger and network vendor Cisco conducted unrelated but similar experiments to measure how often an SSH server accessible from the Internet would be attacked. In both cases a honeypot masquerading as an SSH server was set up. The honeypot would record the usernames and passwords from login attempts. The servers were found almost immediately and logins were attempted. The blogger's experiment lasted six weeks and it collected over 60,000 attempted SSH logins. Cisco ran their experiment for 6 months and recorded 1.56 million login attempts. The results of the research gives credence that people are looking for SSH servers to launch brute force login attempts against. In the case of Cisco's honeypot it averaged having a login attempt about every ten seconds.

Attacking SSH keys isn't just theoretical. There have been a number of publicized incidents in recent years. Below is information on some specific attacks, malware and breaches associated with the loss or use of compromised SSH keys:

- In November 2012 it was reported by the FreeBSD Project that an SSH key was stolen from a developer who had legitimate access to the system resources. The FreeBSD intrusion is a perfect illustration of the dangers of automatic SSH public key authentication without some additional controls or credential verification.
- In January 2013, information circulating on Twitter revealed that users had been caught (via search tools) storing keys and passwords in public repositories on GitHub (a Web-based hosting service for software development projects). Most of the data exposed was personal, but included were private SSH keys. One reportedly exposed SSH password was to a production server of a "major, MAJOR Web site in China."

- A former system administrator for Web hosting company HostGator was arrested in April 2013 for crimes committed in February 2012 after the employee was terminated. The former employee was accused of gaining root access to over 2,700 Webservers by obtaining a HostGator's SSH key and transferring it to computers under his control.
- One of the most interesting SSH key failures occurred in February 2013 when a television station in Montana interrupted its scheduled programming with an Emergency Alert System (EAS) message that announced the beginning of a zombie apocalypse. As a result of this prank it was eventually discovered that the EAS application servers had been shipped with a root SSH private key that had been compromised. With knowledge of that key, hackers accessed the system and broadcast the message they wanted to. The compromised key wasn't disabled until an April update.
- The Ebury SSH Rootkit was first discovered in February 2013 but wasn't widely discussed until April 2014 when it was connected to an anti-cybercrime operation called Windigo. Ebury is a backdoor trojan that is installed on root-level compromised hosts by either replacing SSH related binaries or modifying files used by SSH. Once infected the Ebury malware steals SSH usernames and passwords from incoming and outgoing SSH connections. Private keys stored on the compromised system are also stolen. As mentioned above, once a server is infected with Ebury it has a domino effect as it steals other SSH keys, thus allowing hackers to spread the rootkit. The cybercriminals would use the infected servers to send spam messages and redirect Web traffic. From the information gained from Operation Windigo it was determined that over 25,000 servers from over 60 countries were infected with Ebury. The threat of Ebury has been reduced, but it has not been eliminated.
- The source of the Sony Pictures hack that started in November 2014 has not been determined but it has been reported that within the data the cybervandals posted online included SSH keys. Interestingly on December 7, 2014 the Sony Playstation Network experienced an outage as a result of an attack. Many commentators questioned whether hackers gained access to SSH keys that might have been shared with Sony Pictures personnel who may have had access to both networks.

Mitigating a Point of Sale Threat with SSH

Over the past 18 months there has been a large number of data breaches at retailers and restaurants that have resulted in the exposure of hundreds of millions of credit card numbers. Most of these data breaches were the result of the successful installation of malware onto point-of-sale devices. Backoff is the name attributed to one of the most notorious and prevalent malware applications. This malware is installed after hackers brute force their way into machines using remote desktop applications. The U.S. Department of Homeland Security issued an advisory about this attack vector in July 2014. In addition to providing information about the attack the department also provided suggestions on how to foil attackers. One of the suggestions was to "add an extra layer of authentication and encryption by tunneling your Remote Desktop" through one of the secure tunneling protocols such as SSH.

The Compliance Issue

The requirement to comply with government and industry-specific policies and regulations has been a driving factor within IT security for years. In some areas, regulatory compliance is a dominant factor for IT security. IDC estimates that 80% of the spending on identity and access management is driven by compliance. Regulations are primarily centered on data confidentiality and access controls. The regulatory requirements continue to grow.

Fines and penalties for non-compliance or as a result of a data breach is a costly proposition. Results from IDC's Cyber Threat Survey conducted in May 2014 showed that 80% of financial services organizations and 60% of retail companies were assessed at least one fine over the past two years as a result of a successful cyberattack. In most cases the fines weren't a onetime event with many respondents in both the financial and retail sector being fined at least 25% of the time they suffered a breach. The survey also showed that the fines were not insignificant. Nearly half of the penalties or fines in the financial vertical were at least $100,000 and seven percent of the firms reported they paid over $1 million in penalties.

Organizations, especially in the financial and retail sectors, have to deal with a large set of regulations that have a number of requirements associated with mechanisms required to secure networks and protect data. This paper only highlights the regulatory environment specifically tied to access and data protection. The weakness of many government regulations that are created to enforce laws such as the Sarbanes-Oxley (SOX) Act, Gramm-Leach-Bliley (GBL) Act, and Dodd-Frank Wall Street Reform and Consumer Protection Act are not very specific about how information technology is to be utilized to comply. However auditors have stepped in and suggest that enacting security best practices enumerated by such standards as COBIT (Control Objectives for Information and related Technology) and ISO 27001 will satisfy regulatory compliance requirements.

Related to access controls government regulations prescribe that the following must be achieved:

- Provisioning of access credentials must be documented and approved
- Restrict access to only those specifically authorized to handle specific data
- Track and monitor all access to system resources
- Identify system access by individual user
- Regular recertification of access authorizations
- Termination access credentials when a person changes roles or leaves the organization
- Procedures for creating, changing, and safeguarding passwords
- Provide audit information about who actually accessed, created, modified, or deleted information

Data protection is also a key requirement of government regulations. In order to comply with the regulations auditors require the following:

- Secure personal data from loss and unauthorized disclosure
- Provide proof of protection of regulated data
- Create cryptographic keys in a secure manner
- Protect, rotate and revoke cryptographic keys

In addition to the mandated force of law compliance requirements, many financial institutions and retailers adhere to industry standards such the Payment Card Industry Data Security Standard (PCI/DSS). Unlike the government regulations the PCI/DSS provides very specific activities banks and retailers much follow in protecting payment card information. The section of the standard of most relevance to the use of Secure Shell is Requirement 8: Assign a unique ID to each person with computer access. The requirement states: "Assigning a unique identification (ID) to each person with access ensures that each individual is uniquely accountable for their actions. When such accountability is in place, actions taken on critical data and systems are performed by, and can be traced to, known and authorized users and processes."

Improving Secure Shell Operations

One of the primary motivators for the SSH protocol was the need to solve the problem of user passwords being sent over the network in the clear. Another problem SSH addressed was verifying that the host asking for a password is really that host and not an imposter. SSH does a good job solving these problems by encrypting the connection and verifying the identity of the host and user using a key pair. On the downside, the protocol itself doesn't enforce a strict key management schema. When SSH was in its infancy, many organizations didn't establish the management controls which SSH keys require. Initially, this wasn't a huge problem, but now, with SSH deployments having thousands of keys and millions of associations spread across hundreds of servers, it has become a security and compliance issue. Enterprises have been sloppy about SSH key management, especially when dealing with old, unused, or orphaned keys. Not deleting or changing SSH keys violates the PCI requirements, which can be a huge problem for retailers.

Key management is a difficult task and is generally a complex process. It has been aggravated by the recent resurgence in encryption — driven by regulatory compliance, cloud computing, mobility, and fundamental security needs. To deal with the risk and compliance issues associated with SSH keys, entities must take control of their SSH key management. They must adopt mechanisms that will handle the production, storage, and destruction of keys. Specific actions should include:

- Take inventory of all keys and key associations
- Manage key associations to ensure that access to a specific system by a specific user or process is within policy
- Document what each key is used for, why it was created, and who owns the private key
- Periodically delete old SSH keys and replace them with new keys
- Centralize and restrict authorization to create keys and key associations
- Deprovision keys and key associations when users terminate or access is no longer required

The SSH protocol contains many components that support strong key management, but these components are rarely used in practice because they need to be implemented manually and maintaining them in a dynamic IT environment is difficult. However, there is now a solution that centralizes SSH key management to improve security, compliance, and usability.

Universal SSH Key Manager

Universal SSH Key Manager is a scalable management system that brings SSH infrastructure under identity and access management control. It can be used with either Tectia SSH or OpenSSH, or both. It was developed SSH Communications Security for its customers with three primary objectives in mind:

- Vastly reduce internal and external security risks associated with unauthorized access
- Reduce operational costs by automating much of the complex, manual work required to manage enterprise SSH environments
- Meet or exceed compliance mandates by delivering verifiable access controls, audit, visibility, and reporting

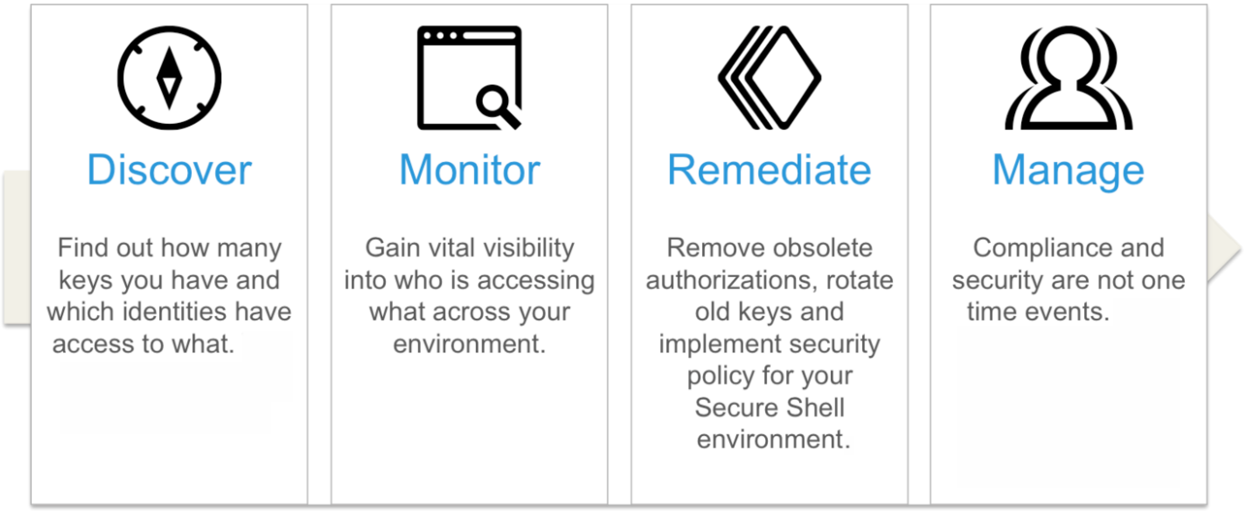
Universal SSH Key Manager delivers effective SSH key management in three core areas:

- **Discovery.** Universal SSH Key Manager has the ability to see into an environment to discover both public and private keys, key size, and key type. The system also automatically identifies whether the key uses a passphrase. It will find rogue and orphaned keys and determine the trust relationship between keys, including unauthorized trust relationships.
- **Central management.** Universal SSH Key Manager acts on the information found during discovery by creating, deploying, removing, and rotating keys through a centralized solution. It allows the administrator to deploy new public keys, create new private and public keys, remove and rotate public keys, and manage one-to-one and one-to-many relationships. Key usage can be restricted by command types, and trust relationships can be restricted according to source and destination. All of this information can be integrated into the IT ticketing system. Additionally, the system allows for the cleanup of the key environment by removing unused keys. A central key store for public keys is maintained for easy recovery. The Universal SSH Key Manager's database, created as a result of key discovery, provides copies of all the network's public keys, fingerprints of the private keys, key associations, and standard information on the keys, such as when they were created, who owns them, and where they are located.
- **Monitoring and reporting.** From a compliance and reporting standpoint, Universal SSH Key Manager provides automated reporting for key, host, and user discovery and key removal/creation/rotation. It provides key activity monitoring, "out of policy" alerts, deep audit trail capabilities, and integration into SIEM/log management systems. It centralizes management of SSH client and server configuration, detects configuration changes, and reverses any unauthorized changes made by end users or malware.

Financial institutions and retailers need to overcome the fundamental barrier of ensuring proper governance in their SSH environment without disrupting ongoing business processes. In addition to security for interactive users, SSH provides a security layer for many automated, server-to-server applications. Changing these applications often entails a level of risk and expense that few enterprises are willing to accept. A significant advantage of Universal SSH Key Manager is it does not disrupt the existing SSH deployment. It complements existing systems and does not require application changes. Figure 2 provides a graphic representation of the functions performed by Universal SSH Key Manager.

FIGURE 2

Universal SSH Key Manager Functions



Discover

Find out how many keys you have and which identities have access to what.

Monitor

Gain vital visibility into who is accessing what across your environment.

Remediate

Remove obsolete authorizations, rotate old keys and implement security policy for your Secure Shell environment.

Manage

Compliance and security are not one time events.

Source: SSH Communications, 2015

With regard to the idea of regulatory compliance, Universal SSH Key Manager meets many of the requirements for data security, key management, identity and access and control. Table 1 provides detailed information about which activities are scrutinized by auditors checking for compliance with regulations and standards.

TABLE 1

How the Universal SSH Key Manager Satisfies Specific Audit Requirements

Compliance Requirement	Status	Potential Audit Finding
Monetary Authority of Singapore	Updated to include specific language concerning cryptographic key management (Appendix C)	Yes
PCI DSS	Access control requirements extended to Secure Shell and other methods of authentication (see SSH website for video)	Yes
NIST-FISMA	NIST IR 7966 "Security of Automated Access Management Using Secure Shell (SSH)" published	Yes
SOX	DS 5.8 – Access control and key management requirements	Yes
NERC	R5 – Account management	Yes
HIPAA	4.x information access management	Yes

Source: SSH Communications, 2015

About SSH Communications

SSH Communications Security was founded in 1995 by Tatu Ylönen, the original inventor of the SSH protocol. The company has focused solely on the development and improvement of SSH for the enterprise and public sector. SSH Communications has over 3,000 customers, including seven of the Fortune 10 and 40% of the Fortune 500. The company is headquartered in Helsinki, Finland, and maintains regional offices in Hong Kong, Germany, Switzerland, and the United States. The company is listed on the Helsinki Stock Exchange under the stock symbol SSH1V.

THE CHALLENGE OF AWARENESS

SSH is a tool primarily used by systems administrators, application developers, and automated server-to-server applications. Many IT professionals who work with SSH are aware of the security and operational problems associated with large SSH deployments. However, these issues have largely "flown under the radar" of the IT management ranks, including many security professionals. Personnel working in the trenches are often simply living with these issues because the managers who are providing direction view SSH simply as a protocol (if it is considered at all).

IDC believes this scenario is evolving and the rate of change is likely to accelerate. A primary driver of accelerating awareness is the tightening of security standards to include the need for governance of SSH authorizations. In some cases, this is a matter of reinterpreting existing standards language — recognizing that an SSH key is functionally equivalent to a username and a password. In other cases, the standards are being updated to explicitly include key management.

Fortunately, organizations that bring their SSH environment into compliance through comprehensive central management can realize the added benefits of reduced risk and reduced cost. Reducing the overhead of a manually operated SSH infrastructure results in a quantifiable and significant ROI, which is not the case with many other compliance-driven security investments. This might be one time when IT managers and business executives can drive the deployment of an IT technology that improves administrators' productivity, is cost effective, improves security, and supports compliance.

CLOSING COMMENTS: ARE YOUR SSH KEYS WELL MANAGED?

The SSH protocol is an IT tool that is widely deployed without any fanfare. It is part of the IT "plumbing," used by IT administrators and professionals to establish encrypted connections. Nearly every enterprise is using SSH, even if the management doesn't know about it. In most cases, the tools used by the IT department are of little concern to management; however, SSH is more than a simple tool. Through SSH, it is possible to gain considerable access to the network. For this reason, SSH requires special attention. The specific threats have been discussed previously. IDC has high confidence that every enterprise without central SSH key management is exposed to the risks and issues outlined in this white paper.

The first step in understanding the scope of the problem is to take inventory (via network discovery) of the authorized keys on SSH servers and user identities on SSH clients. With SSH following a self-service key management model, IDC is certain that enterprises will find a high number of keys and that a significant percentage of these existing SSH keys will be unmatched. That is, a public key will be without a corresponding private key, or alternatively, a private key won't have matching public keys. Solving this problem alone is worth the investment in a central key management solution. However, there are considerable advantages to using a key manager, such as Universal SSH Key Manager from SSH Communications Security, which goes beyond the discovery of SSH keys. Central key management returns control for the creation and removal of keys. It provides a backup/recovery capability as well as user access information for auditors.

There is a general enterprise requirement to bring IAM governance to SSH environments. Moreover, as SSH deployments increase in tandem with the widespread expansion of cloud and virtualization technologies, the need grows ever greater. IDC believes every enterprise, especially financial institutions and retailers, should include SSH governance as a critical component of their IAM strategy for security, compliance, and cost reduction.

About IDC

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. IDC helps IT professionals, business executives, and the investment community make fact-based decisions on technology purchases and business strategy. More than 1,100 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For 50 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world's leading technology media, research, and events company.

Global Headquarters

5 Speen Street
Framingham, MA 01701
USA
508.872.8200
Twitter: @IDC
idc-insights-community.com
www.idc.com