SECTIGO®

sectigo.com

WHITE PAPER

Why Certificates Are a Better Approach to SSH

SSH keys leave you exposed – certificates reduce costs and institute greater control

What Is SSH

SSH (Secure Shell) is a network protocol that allows one device to securely connect and communicate with another device or server. In use for more than 20 years, SSH is a popular technology that is integrated into many different operating systems in some shape or form. Indeed, increased investment in DevOps and cloud-based solutions has resulted in a proliferation of servers and therefore of SSH operations.

This paper will discuss the considerable benefits of certificated-based SSH authentication, a secure and cost-effective approach to SSH.

Increasing cybersecurity concerns require enhanced control over administration, and certificate-based SSH delivers substantial improvements to the SSH community. Sectigo believes that certificated-based SSH offers organizations significant cost-savings along with increased fine-grained controls over administrator roles.

Types of SSH

There are a few different applications of SSH technology used today.

One implementation, OpenSSH, is integrated into Windows, Linux, and other major operating systems. It has grown in adoption and is ubiquitous in securely accessing remote systems.

SSH authentication can also be accomplished through a username/password approach. However, as typical with any password-based system, it has been shown to be insecure and should not be used today.

The most popular SSH authentication technique utilizes a key-based authentication format, where a private/public key pair is generated through cryptography. It is accomplished by installing the public key into each server and keeping the private key secret in the client. This technique allows users to remote host into different systems, such as servers, workstations, or other physically distant devices, using root permissions to log in.

The final option is certificate-based SSH authentication, which is increasingly recommended by industry experts as the best practice for SSH authentication. Key-based and certificate-based SSH authentication use the same cryptographic techniques, but certificates offer several other benefits over keys alone, especially in the areas of granularity and costs.

Certificate-based SSH

Certificate-based SSH authentication emerged as an option of OpenSSH in 2010. An SSH administrator certificate is a data structure that binds the SSH public key to the administrator's name, expiration date and permissions. The data structure is cryptographically signed by a certificate authority (CA), an entity trusted by the SSH client and server. This means that each server does not need to maintain their own repository of files mapping SSH public keys to administrators, and there is no need to erase the public key file to remove access. It simply expires or is revoked by the same CA that signed the certificate. The server uses cryptography to validate the certificate is authentic and unaltered. Instead of scattering public keys across static files, a public key is bound to a name with a certificate. SSH certificates can be cryptographically verified and, like traditional SSH keys, are exchanged between client and host during the SSH handshake. These certificates present an initial benefit within the SSH handshake as they do not trigger a warning on first use. Such warnings are usually skipped over by users during the handshake initialization and, although harmless in this scenario, ignoring them creates bad security habits. These habits trigger unsafe behaviors during events like unsecured website warnings or unfamiliar file warnings, both of which contain huge amounts of risk for an organization.

Certificates offer the best method for SSH authentication. Their adoption has grown over the past few years within large, tech-based organizations where they are employed internally. Certificate use will continue to grow, as they apply several benefits of traditional key-based SSH while shoring up many of its faults. The ability to authenticate with SSH certificates is already built into the OpenSSH client and software, there is no additional software investment required.

Considerations When Using SSH Keys

SSH using key-based authentication carries inherent complications that increase operations costs.

01

Costly key tracking

With SSH keys, an organization must install each IT administrator's public key to every server they intend to access, resulting in a complicated web of keys across the organization. To keep track of all keys, organizations spend countless man-hours completing this process manually or purchasing third-party software to assist. Depending on the size of the organization, these costs can be prohibitive.

02

Lack of scale

Additionally, SSH keys have a scaling issue. When the number of SSH keys balloons, admins have a real problem on their hands in the form of key sprawl. Cleaning up scattered, discarded keys also costs SSH operators significant amounts of time and manpower.
Of course, IT admins should delete their public key file from the server when they no longer need access, but this rarely happens in reality. Many platforms offer costly SSH key discovery as an add-on, which has become fairly accurate but, ultimately, will not find every key stored within an organization. Unfortunately, it only takes one pair of compromised SSH keys to completely expose a system.

03

Loss of control

SSH private keys can be stored directly in a client's machine or in a smart card/USB token.
If the private key is extracted from the machine or the smart card/USB token is stolen, then a malicious actor can impersonate the administrator. This represents a risk for the enterprise, or at minimum a costly effort to delete the matching public keys from all the servers as quickly as possible. If not erased, the resulting breach would open the organization to reputational damage and regulatory fines.

04

Unlimited validity

Furthermore, these keys were not created in a way that allowed for a validity period. Essentially, the keys are valid forever, allowing someone who used them legitimately once for a specific purpose to continue to access the system once their original purpose has passed. With SSH keys, an organization must go to every associated web server and delete the public keys when someone should no longer have access. Similar to key sprawl, solutions to this issue can come in the form of expensive tools or through many man-hours to revoke access manually.

The open-source community has created some SSH key revocation solutions, but they have not been widely adopted. These solutions are cumbersome, hard to customize, and face major useability issues. In practice, it is apt to say that SSH keys are permanent unless specifically rekeyed.

Key-based SSH Breaches

The inherent risks in key-based SSH authentication may sound theoretical, but they are not. Countless real-world examples of bad actors utilizing key-based SSH for malevolent purposes exist, such as the two examples below of well-known data breaches.

Sony Breach

In one of the most famous data breaches of all time, Sony Pictures Entertainment was breached by a hacker group identifying themselves as Guardians of Peace. These hackers used stolen SSH keys to gain unauthorized access to a single server, which allowed them to further infiltrate the Sony systems, gaining numerous additional SSL private keys while leaving backdoors along the way. The breach was disastrous to Sony and included unreleased IP information, personal identifying information (PII), and even the SSH keys to the Sony payroll system. The impact was so large it forced Sony to bring its entire network offline including its VPN and employee workstations.

To secure its systems, Sony needed to rekey its private keys and understand how each of the keys within the breach were used. This takes a significant amount of time and man-hours. If Sony had used SSH certificates instead of keys, it would be a matter of certificate revocation and require much less effort, while reducing the costs to Sony. Sony's data security reputation has been harmed ever since. Hosted in numerous places, data from this breach can still be found, and is often used by security researchers as an example of real-world data to train their models. Once data is stolen, it remains available forever.

GoDaddy Breach

In 2019, GoDaddy, a large provider of domain names, reported a breach within their systems that allowed unauthorized access to some of their SSH accounts. Initially, a small number of affected users were reported but it was later discovered that at least 28,000 users were involved. This breach was critical not only because a large number of users were affected but also due to the length of time the attackers were in GoDaddy's hosting environment and network, approximately six months.

This amount of time allowed the hackers to understand their scope of access and plan further attacks on customers that utilized GoDaddy SSH keys. Although there is no evidence that the hackers were able to successfully utilize this access, the capacity for damage with access to the SSH keys of over 28,000 different accounts is no small matter. It would take an incalculable amount of time to fully rekey and protect organizations from every accessible account. If the breached had involved compromised certificates, it would also be a serious situation; but there would be the certainty that the SSH certificates both have a validity period and a process to be revoked.

Developing a strong security program often involves a planned and deliberate patching strategy and process, thorough testing, and active monitoring of critical systems; all of which are easier if your SSH practice relies upon certificate-based rather than key-based authentication. In both previous examples, and countless more, usage of SSH certificates over SSH keys would have saved the organizations significant amounts of time and money by offering increased control over their utilization and revocation.

Why Certificate-based SSH Is the Solution

Modern SSH using certificate-based authentication is the most cost-effective method of providing access to authorized users while maintaining the cryptographic security of key-based authentication.

Allowing for more centralized control, certificate-based authentication has several benefits and comes in multiple formats depending on the use case. It's important to note that even though creating a SSH certificate takes about the same labor as creating a SSH key, the increased benefits it offers to an organization are unmatched.

Single pane of glass manageability

SSH certificates are easier to manage from end-to-end, from issuance to revocation. The certificate rollout is simpler than a key rollout, especially using the Sectigo Certificate Manager platform. SSH Certificates can be managed from the same system as all other certificates in the enterprise, consolidating administrative operations in a single UI, simplifying management and reducing the man hours required to accomplish tasks.

When an organization is using SSH certificates, tracking of the authenticity of access is done by the SSH server. This is achieved by the system simply asking the administrator's desktop for the corresponding SSH certificate during the login process – automatically, eliminating the need for any expensive key tracking tools required with SSH keys.

Controlled validity periods

SSH certificates will expire when they are no longer needed, since their lifespan can be customized. They can also be revoked, unlike SSH keys, in case the access needs to be ended before expiration date.

Due to these validity periods, certificates with shorter lifespans have become more popular with administrators. Certificates offer increased security benefits as the shorter the validity period the more secure the SSH certificates are considered. Certificates administrators can set the validity period as short as hours, or even minutes, depending on their planned use.

SSH certificates also have the capability to limit, via a field on the SSH certificate, what the administrator can do on the server during that login. This, coupled with short-lived certificates offers supremely fine-grained control over SSH keys within an organization.

Reduced labor requirements

The cost of managing many keys across multiple servers is prohibitive. Certificates introduce a cost-effective, well managed approach. In the example below, we can see how SSH certificates have the potential to save an organization thousands of man-hours while creating a more streamlined environment. In contrast, SSH keys is the tedious process of pushing the user's public key to each server. Let's assume that 30 users want to login to 500 servers:

Using SSH key authentication to activate users:

30 users x 500 servers x 5 min. per key copy

= 75,000 minutes spent

Using SSH certificate authentication to activate users:

500 users x 1 minute per cert creation

= 500 minutes spent

Since certificate-based authentication has validity periods, even if an employee has left the company or the private key is lost, administrators can be confident that after a determined period the risk associated with the certificate will be gone. This also applies at scale. When administrators leave organizations, the loss of institutional knowledge including what keys exist and where they are located can be devastating. SSH certificates mitigate this risk by issuing a new certificate for every new administrator, reducing the time and resources spent on onboarding necessary new administrators.

Again, let's quantify the effect that SSH certificates have on eliminating man-hours, this time using employee departures throughout a year. Let's now assume that in the previous group of 30 people using SSH for 500 servers there's a 10% attrition rate:

Using SSH key authentication for deauthorizing users:

30 people x 10% attrition x 5 minutes to delete each key in 500 servers

= 7,500 minutes per year

Using SSH certificate authentication for deauthorizing users:

30 people x 10% attrition x 5 minute to revoke the certificates

(through automated action if the revocation system is linked through an organization's active directory or HR system)

= 15 minutes per year

With SSH certificates, when access is removed from one system, it is removed from all systems. The cost savings in man-hours alone makes the SSH certificate implementation worthwhile in the long run.

Simple revoke and replace

Additionally, SSH certificates ease the inconvenient practice of rekeying. As previously mentioned, keys are not always the best protected since they are stored in known locations. Therefore, best security practices would recommend regular rekeying, especially after a breach or security incident. This is not always possible in key-based authentication since it requires the administrator to know the location of every key. With SSH certificates, revocation is part of an established framework. Changing certificates is as easy as pressing a few buttons within the Sectigo platform.

While SSH key management tools can be purchased to delete unused keys, an SSH certificate offers the additional benefit of automatic termination of a person's authentication credential, saving organizations unnecessary expenses. Single pane of glass management makes this simple and reduces the human error of forgetting to remove access and the need to delete public keys in hundreds of servers.

Summary

Although key-based SSH authentication is the more popular form of SSH authentication today, it has serious drawbacks in areas such as:

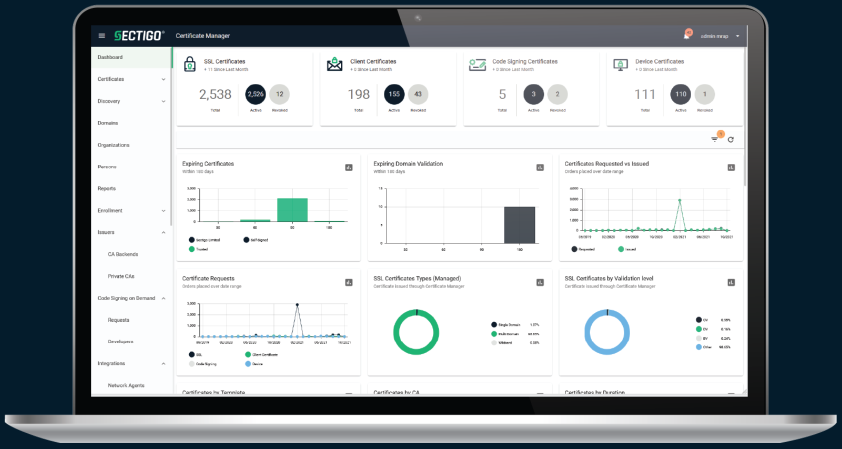
- **Costly key tracking**
- **Lack of scale**
- **Loss of control**
- **Unlimited validity**

Certificate-based SSH authentication overcomes these limitations, that ultimately result in significant costs to the business and risk considerable reputational damage. Certificate-based SSH has the following benefits:

- **Single pane of glass manageability**
- **Reduced labor requirements**
- **Reduced costs**
- **Controlled validity periods**
- **Simple revoke and replace process**

Investing in certificate-based authentication will save any organization from numerous headaches and pitfalls, especially when working with an experienced vendor that fully leverages the benefits of digital certificate authentication.

Sectigo Certificate Manager



Sectigo Certificate Manager (SCM) is a fully automated certificate management solution purpose-built for today's enterprise environment. It provides complete visibility and lifecycle control over all public and private certificates and keys from a single platform. In addition to providing tools for automation of client and server certificates, SCM provides management tools for code signing, document signing and SSH certificates. As certificates are increasingly recognized as being the foundation for secure communication between systems, a single platform to manage these digital identities is an essential tool in the modern enterprise. SSH Certificates are an example of how SCM provides immediate and compelling value to the enterprise.

With SCM you can secure your human and machine identities at scale with flexible deployment using integrations into all leading technology providers.

About Sectigo

As a trusted public Certificate Authority, a founding member of CA / Browser Forum, and issuer of hundreds of millions of certificates, Sectigo is the expert on all things to do with certificates.

We protect private roots of our SSH certificates at the same level applied to the hundreds of millions of public digital certificates we've issued worldwide. Our legacy and tradition ensure that enterprise security programs remain at the leading edge of cryptography as your needs change and grow. We are on the forefront of SSH security, being the only major public CA to offer SSH certificates.

Sectigo enables today's enterprises to securely conduct business in the digital-first world. Trusted by more than 700,000 customers, Sectigo partners with organizations of all sizes to deliver best-in-class automated Certificate Lifecycle Management. Sectigo's cloud-based platform delivers speed, flexibility, and scale to those securing today's most complex enterprise environments, while future-proofing businesses against threats. With 20+ years of industry experience as the world's largest certificate authority (CA) and more than 400 million public certificates issued, Sectigo is the leading Certificate Lifecycle Management provider supporting multiple CA vendors and integrating with the largest software ecosystems in the world. Learn more at www.sectigo.com and @SectigoHQ