How Uber, Facebook, and Netflix Do SSH

JUL 16, 2019 BY JON SILVERS

According to one study, there is a cyber security attack every 39 seconds. Defending against such attacks has become of paramount importance to how businesses operate today. The traditional approach to securing IT infrastructure is perimeter-oriented solutions such as firewalls, VPNs, and password policies. While these are all good and necessary, they don't speak to the security vectors that are "built in" to an organization at the code or infrastructure level. This is perhaps why we're seeing more security fall into non-traditional areas such as compliance, devops, or engineering.

Privileged access management for cloud-native infrastructure that doesn't get in the way. Learn More...

Kubernetes packaging solution that takes the drama out of on-prem deployments. Learn More...

ARTICLES BY TOPIC

On the devops level, where many of our customers and end-users are working, more

Choose a topic

teams are putting their focus on protecting infrastructure, and often that begins and ends with Secure Shell (SSH) network protocol. SSH is used to securely access remote machines, and for authentication and encrypted data communication. Under a typical server-client SSH setup, the server trusts the client if the client's public key is in a list of authorized public keys, and the client trusts the server's public key upon the first encounter.

There are some flaws with this basic setup. First, authorized_keys files can become large and unmanageable. Next, many SSH clients allow trust on first use (TOFU), which offloads the decision making to the user on what to trust. Many users will bypass the messages which will allow connections to untrusted hosts and potentially leave machines vulnerable to man-in-the-middle attacks. A solution to these problems is to move from using public keys to using SSH certificates and certificate authorities (CA) as the trust mechanism. Using SSH certificates, administrators can set specific rules for obtaining them and defining expiration rules.

Some of the largest companies in the world are working through SSH flaws by taking major steps like not issuing keys or taking a "Zero Trust" approach to authenticating and authorizing network users or devices. The term, coined by Forrester, refers to flawed trust assumptions security leaders make about internal network traffic, employees, and 3rd parties who may have access to systems.

One of the most well-known examples of a Zero Trust model is Google's BeyondCorp model. While many other companies have been adding layers of firewalls, restrictions, and VPNs to gain access to systems, Google has done the opposite. No longer does the company rely on a network perimeter, but rather exposes internal systems to the public internet. This requires all systems employed within the BeyondCorp model to be built with the same skill and hardening as required by any public internet solution.

As a company who sells a modern, cloud-native alternative solution to SSH bastions to address infrastructure-related security, we asked ourselves, how are some bigger, well-known companies approaching SSH? We took a look at three companies who are setting an example for others to follow and who have written publicly how they're approaching SSH – Uber, Facebook, and Netflix.

Uber

The Uber security team saw several problems with SSH public/private keypair model. In addition to the key management problem mentioned earlier, there was no automated process to expire keys. The longer that keys are valid, the greater the risk the key is to be lost or stolen.

The team was also concerned about the excessive number of two-factor authentication (2FA) requests employees received. Constant prompting for 2FA would lead to fatigue and employees might start accepting any kind of 2FA request.

Most of these issues were resolved when Uber started using SSH certificates with OpenSSH 5.4, but the organization still felt there was no current solution that met their unique needs. Specifically, they needed support issuing both user and host certificates, and they wanted to be able to continuously authenticate a user instead of just at a single point in time.

They developed the Uber SSH Certificate Authority (USSHCA) along with a pam module for continued validity of a user. Uber employees are issued SSH certificates by USSHCA. Each certificate has a lifespan and can be configured based on the individual employee role or their group within the company.

Facebook

Facebook's security team uses certificates instead of public key authentication for the following reasons:

- Local account management will get unruly as the company continues to grow.

- Central authentication is a single and dangerous point of failure.
- It's almost impossible to scale trust of individual key-pairs.

In developing a system that met the company's needs, Facebook borrowed a widely-used concept used for HTTPS traffic. The company configured its SSH servers to trust their CA and everything that it signs. The next step is authorization; the certificate contains all access and privileges for specific employees. Thus, everyone has the exact amount of access that they require.

Facebook also uses OpenSSH to collect logs in real-time on certificates that were used for authentication, a necessary task for compliance and accountability.

Netflix

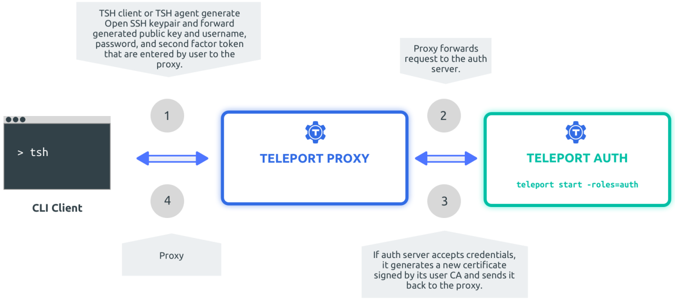
Netflix's security team developed its BLESS (Bastion's Lambda Ephemeral SSH Service) certificate authority to align security with its engineering culture of "freedom and responsibility." Engineers at the company are expected to:

- Proactively and openly share information
- Provide context as to why an action is performed
- Operate all services that their team creates

As a result, almost every engineer at Netflix needs SSH. And they need to be able to access their services with as little friction as possible.

BLESS runs on AWS Lambda and uses Amazon's Key Management Service for encryption. Netflix's SSH bastion architecture reduces friction by using SSO to authenticate users and issuing short-lived certificates. The company protects itself by running automatic scanning of SSH usage and sounding alerts if something suspicious is suspected.

An Accessible Zero Trust Approach for Infrastructure



TSH client or TSH agent generate Open SSH keypair and forward generated public key and username, password, and second factor token that are entered by user to the proxy.

Proxy forwards request to the auth server.

If auth server accepts credentials, it generates a new certificate signed by its user CA and sends it back to the proxy.

```
> tsh
```

```
teleport start -roles=auth
```

Forrester says that companies must implement Zero Trust security principles to "future-proof" their businesses and protect themselves against advanced cyber attacks. But where do you begin? Do you take Google's approach and develop an entirely new security model? Or you could take Facebook's, Uber's, or Netflix's approach of internally developing your own safeguards. While rolling your own certificate authority solution is ideal, it's still out of scope for many organizations who want to stay focused on their core business.

The good news is that you don’t have to have a Google-sized budget to afford or implement a certificate authority approach. Teleport works with your existing SSH to provide role-based access controls. With Teleport, security teams don’t have to worry about managing keys, firewalls, or VPNs. If you’re interested to explore more, you can download our free, open source edition of Teleport to see how it works in your environment. Or, schedule a demo and we can walk through Teleport with your team.

ssh bastion certificate authority

Twitter Facebook LinkedIn

Want to stay informed?

Subscribe to our weekly newsletter for the latest articles, industry changes, and products updates.

Email Address SIGN UP

Connect with Us

May 9, 2019

SSH Handshake Explained

By Russell Jones

Apr 15, 2019

Teleport Helps Auth0 Meet PCI and SSH Requirements

By Jon Silvers

Apr 1, 2019

Set up a SSH+Kubernetes bastion for AWS EKS with Teleport 3.2

By Ev Kontsevoy

Start Using Teleport Today

Teleport gives you security best-practices out of the box for the privileged access management of your cloud-native infrastructure.

DEMO TELEPORT

DOWNLOAD TELEPORT

PRODUCTS

Teleport

Gravity

Teleconsole

LEARNING

Resources

Teleport

Docs

Gravity

Docs

COMPANY

About Us

Careers

The Blog

Press

Coverage

GET IN TOUCH

Customer Support

info@gravitational.com

Phone: (855) 818 9008

CONNECT

Community

Forum

Github

Twitter

© Gravitational, Inc. All Rights Reserved

Terms of Service

Privacy Policy