EZSSH
SSH Made Easy!
Cloud adoption is making companies move to a zero-trust networks.

99% of compromises involve a stolen credential\(^1\)

Stolen SSH credential attacks are on the rise.

Companies are spending millions of dollars on improving their corporate identity.

Linux servers do not use Active Directory Accounts.

Thousands of keys are leaked on GitHub each day.\(^2\)
SSH KEY EXPOSURE: LAPSES IN SERVER ACCESS SECURITY

SSH (Secure Shell) is a secure way to remotely connect to and communicate with servers. But the way in which the keys are handled can lead to access security issues and the potential exposure of sensitive data.

SSH keys are used to log into servers (more secure than just a username and password!). These keys include a public key and private key that are cryptographically secure and used to authenticate a client to an SSH server. The private key should be kept secret and protected. The public key, on the other hand, can be stored online or in a keychain and is necessary for remote access.

Oct 24, 2017
By Ths T. Pham
WHY GO PASSWORD LESS?

Passwords are no longer secure due to brute force attacks.

72% of individuals reuse passwords in their personal life while nearly half (49%) of employees simply change or add a digit or character to their password when updating their company password every 90 days.

Compromised passwords are responsible for 81% of hacking-related breaches, according to the Verizon Data Breach Investigations Report.

Microsoft recently announced that a staggering 44 million accounts were vulnerable to account takeover due to compromised or stolen passwords.
ARE SSH KEYS THE SOLUTION?

- Linux servers in large corporations have between 50 and 200 SSH keys.
  - 90% of those keys are not used.
  - SSH keys never expire.
  - 50-200 keys per server.
- Keys must be manually life cycled.
- No Advance Identity Protection
  - Conditional Access
  - Smart Alerting
  - Just in Time Access
- Hard to keep inventory of which key gives access to who.
- Engineers don't follow best practices to protect the keys.
- Current Linux Systems are protected in two ways:
  - Creating an account for production and sharing the credentials among engineers.
  - Creating accounts for each engineer in each of the servers.
Engineer goes to a site and learns how to create an SSH key.

Engineer creates the key.

Sends it to security team or server admin to be added to the server.
CURRENT WORKFLOW

Each User Gets an Account

Security team adds it to the server.

Engineer gets access to the server and now can start their work.

When engineer no longer needs access, is the account removed?
Engineer goes to a site and learns how to create an SSH key.

Poor key hygiene, no key clean up since it is hard to keep track of who still needs access.

Long and tedious access reviews.

High price to onboard new team member

Key reuse over different scopes.

Keys are not properly protected by users.
CURRENT WORKFLOW

Shared Accounts

Engineer goes to team wiki to get key locations

Engineer gets the key from team shared location

Engineer saves the key in their system

Engineer accesses server and now can start their work

Many of these keys are reused between test and production.
ONE ACCOUNT FOR ALL ENGINEERS

- Usually, keys are shared in unsecure ways such as: email, file shares, wikis, git.
- Hard to rotate since all engineers would have to get the new key.
- When an employee leaves, they can maintain access to servers.
- Big insider threat opportunity (61% of CIS0s worry about insider threats).
- Not possible to know who did which change since all server logs show being done by the same account.
- No approval flows to get access to server
- Reuse of "team keys" for many services.
YOUR CURRENT COST

- Engineering time creating and passing the keys
- Security team time adding and removing keys
- Added risk for having engineers manage key and access
- SSH key inventory and access reviews for all your servers
- Added risk for life cycling accounts out of the environment when no longer needed
SSH CERTIFICATES

- Poorly documented
- Need cryptographic knowledge
- Manual setup and management
- No automatic provisioning
- No approval workflow
- Most of the large companies use it (with custom built tools)
HOW CERTIFICATES WORK

1. Request Access
2. Approval Workflow
3. Request Signed Certificate
4. Return Signed Certificate
5. Return Signed Certificate
6. Access machine with Signed Certificate
Our Solution

- Seamlessly integrates with Azure
  - Works with Azure Security tools such as Azure JIT and Azure PIM.
  - Integrates with Azure RBAC for automatic access management.
  - Automatically adds Azure Servers to your policies.

- Works with hybrid and multi-cloud.

- Easy setup for all your servers.

- Uses your secure corporate account to create time bound certificates.

- Makes security transparent to the user

- Automatically onboards new team members

- Approval workflow for critical environments

- Automatically removes access when no longer needed
EZSSH ADVANTAGES

- Designed for Zero Trust networks
- Reduce Onboarding time and cost by removing need to manage SSH keys
- Remove key management overhead from engineers.
- Reduce insider threat by having Just In Time Access with appropriate approval workflows.
- Reduce audit costs with easy to Audit access logs
- Reduce offboarding time and risk.
- Native Linux Authentication no custom PAM module or code runs on your servers.
- Bring your own Certificate Authority support
DEMO
EZGIT
Protecting GitHub with SSH Certificates
Hackers are targeting developer credentials to steal code.

SSH Keys are not properly managed by users.

SSH Certificates are supported but there is no infrastructure to issue them.

Need Secure infrastructure to run your own Certificate Authority.

Conditional Access does not apply to the most critical operations
GitHub Breaches

GitHub leaks exposed up to 200,000 medical records: 4 details

Report: Microsoft’s GitHub Account Gets Hacked

Compromised SSH keys used to access popular GitHub repositories

Security experts Ben Cox explained that the official Github repositories of the UK Government, Spotify, and Python were accessed using compromised SSH keys.
GitHub is forcing you to go password-less in 2021.
- Gives you an opportunity to modernize your development security stack.

- Reduce surface area with short-term SSH Certificates
- Reduce engineer onboarding time
- Make audits easier with easy to audit logs
- Make security transparent for your users.
Uses your Secure Azure AD Identity for Authentication of your developers.

Seamlessly integrates with our VM offering.

Easy setup with any Git offering.

Uses your secure corporate account to create time bound certificates.

Makes security transparent to the user.

Automatically onboards new team members.

Integrates with any git tool that uses ssh-agent as authentication method.

Automatically removes access when no longer needed.
HISTORY OF SSH AUTHENTICATION

1995 SSH Created

2010 SSH Certificates introduced

2021 EZSSH Makes SSH Certificates easy to use
AZURE USERS NEED A SOLUTION

Posted by u/bluifer_jr 1 month ago

**linux VM ssh key rotation strategies?**

Security

Curious how folks are managing key rotation for linux VMs.

Had hoped to leverage terraform but changing azurerm_linux_virtual_machine.admin_ssh_key value forces a re-create of the VM and associated resources. That'd be fine for most things but I'd like to avoid doing for long-running database VMs.

Can see some other ways like scripting and updating in CLI or powershell, which I suppose I could also run after creating new VM in terraform, just seems like a bad path to start down.
ENGINEERS MAKE MISTAKES

The bulk of insider data breaches

- Employees rushing or making mistakes: 60%
- Lack of awareness: 44%
- Lack of training: 36%
- Intentionally leaking data: 30%
SSH UNDER ATTACK

- [https://blog.ssh.com/ssh-key-scan-attack-honeypot](https://blog.ssh.com/ssh-key-scan-attack-honeypot)
- [https://www.beckershospitalreview.com/cybersecurity/github-leaks-exposed-up-to-200-000-medical-records-4-details.html](https://www.beckershospitalreview.com/cybersecurity/github-leaks-exposed-up-to-200-000-medical-records-4-details.html)
- [https://thehackernews.com/2021/08/how-companies-can-protect-themselves.html](https://thehackernews.com/2021/08/how-companies-can-protect-themselves.html)
2021 had the highest average cost in 17 years

Data breach costs rose from USD 3.86 million to USD 4.24 million, the highest average total cost in the 17-year history of this report.

Remote work due to COVID-19 increased cost

The average cost was USD 1.07 million higher in breaches where remote work was a factor in causing the breach, compared to those where remote work was not a factor.

Compromised credentials caused the most breaches

The most common initial attack vector, compromised credentials, was responsible for 20% of breaches at an average breach cost of USD 4.37 million.

Security AI had the biggest cost-mitigating effect

Automation and security artificial intelligence (AI), when fully deployed, provided the biggest cost mitigation, up to USD 3.81 million less than organizations without it.

A zero trust approach helped reduce cost

The average cost of a breach was USD 1.76 million less at organizations with a mature zero trust approach, compared to organizations without zero trust.

Cloud migration impacted costs and containment

Organizations further along in their cloud modernization strategy contained the breach on average 77 days faster than those in the early stage of their modernization journey.
<table>
<thead>
<tr>
<th>Tool Name</th>
<th>How It Works</th>
<th>Key Drawbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thycotic Secret Server</td>
<td>It is a shared password manager that allows teams to centralize their</td>
<td>- Requires an admin account with password to run as a high privilege</td>
</tr>
<tr>
<td></td>
<td>password manager.</td>
<td>user to rotate the passwords and keys.</td>
</tr>
<tr>
<td>Hashicorp Vault</td>
<td>Hashicorp vault is a vault service that allows you to store and create</td>
<td>- While vault offers SSH Certificates that is the same tech that we use, the</td>
</tr>
<tr>
<td></td>
<td>secrets for your endpoints. It also has an SSH CA feature that allows you</td>
<td>process for the user is still manual (they must go to vault, create the</td>
</tr>
<tr>
<td></td>
<td>to create SSH certificates.</td>
<td>certificate and then install it on their PCs).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Vault also lacks the advance access management that EZSSH offers.</td>
</tr>
<tr>
<td>Key Factor</td>
<td>Key factor allows companies to centralize their SSH key management into one</td>
<td>- Requires admin privileges to manage SSH credentials.</td>
</tr>
<tr>
<td></td>
<td>portal.</td>
<td>- While key management is centralized, input from administrators is required</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for lifecycle credentials.</td>
</tr>
</tbody>
</table>
**UNIQUE EZSSH FEATURES**

- Designed for Zero Trust (No agent or high privilege account)
- Connection with Azure security tools: Networking JIT, Azure PIM, Sentinel
- Transparent security for users, with easy-to-use tools
- Reduce insider threat by having Just In Time Access with appropriate approval workflows.
- Reduce audit costs with easy to Audit access logs
- Centralized management for hybrid and multi-cloud environments
- Native Linux Authentication no custom PAM module or code runs on your servers.
- Bring your own Certificate Authority support