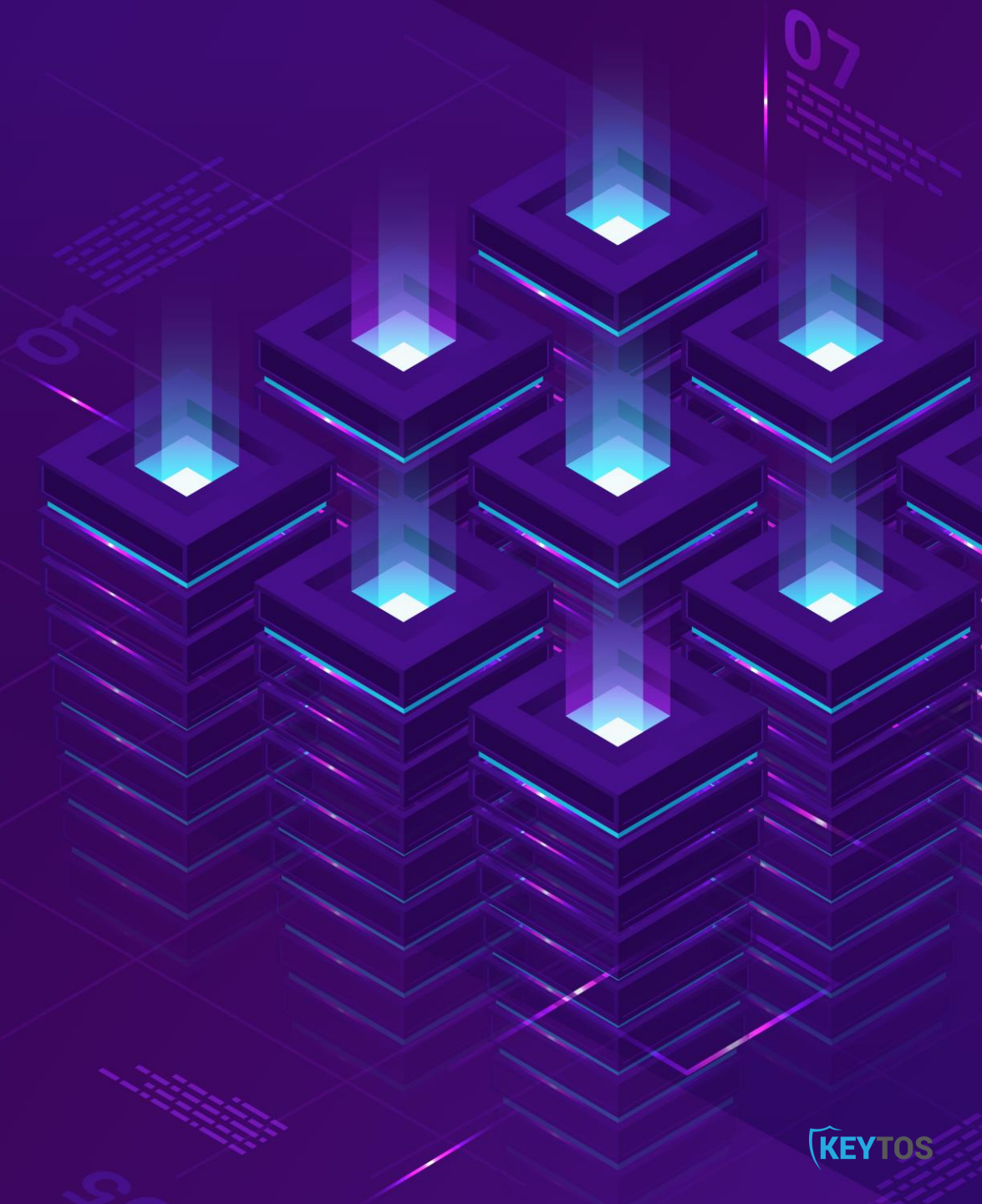
KEYTOS

EZSSH

SSH Made Easy!



KEYTOS

PROBLEM OVERVIEW

The move to the cloud is making companies move to a zero-trust networks.

99% of compromises have use a stolen credential.

SSH stolen credential attacks are on the rise.

Companies are spending millions of dollars on improving their corporate identity (Web Auth, AD, MFA)

Linux servers do not use Active Directory Accounts.

KEYTOS

IN THE NEWS

KEYTOS

WHY WE HAVE TO 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.

KEYTOS

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.
  - Most organizations have 10-50 times more keys than they estimate.
- Keys must be manually life cycled

- 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.

KEYTOS

CURRENT WORKFLOW

Each User Gets an Account

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.

KEYTOS

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?

KEYTOS

EACH ENGINEER CREATES AN ACCOUNT PROBLEMS

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

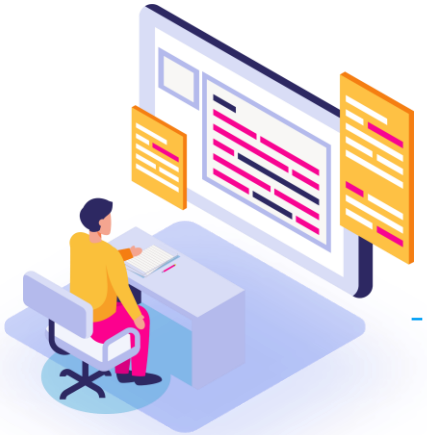
Key reuse over different scopes.

Keys are not properly protected by users.

KEYTOS

CURRENT WORKFLOW

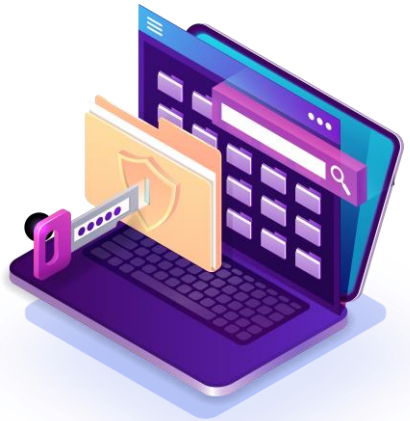
Shared Accounts



Engineer goes to team wiki to get key locations

Many of these keys are reused between test and production.

Engineer gets the key from team shared location



Engineer saves the key in their system

Engineer accesses server and now can start their work

KEYTOS

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 CISOs 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.

KEYTOS

YOUR CURRENT COST

SSH key inventory and access reviews for all your servers

Added risk for life cycling accounts out of the environment when no longer needed

Added risk for having engineers manage key and access

Security team time adding and removing keys

Engineering time creating and passing the keys

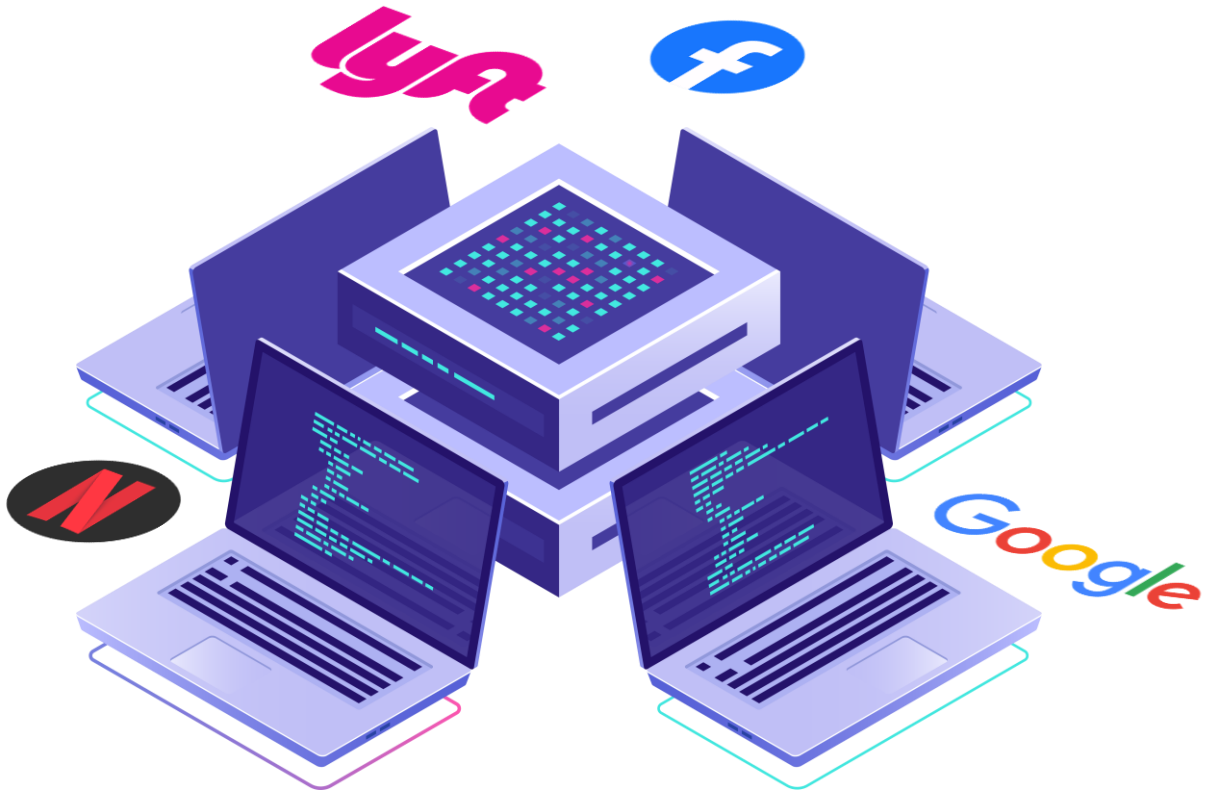
KEYTOS

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)

KEYTOS

HOW CERTIFICATES WORK

1. Request Access

2. Approval Workflow

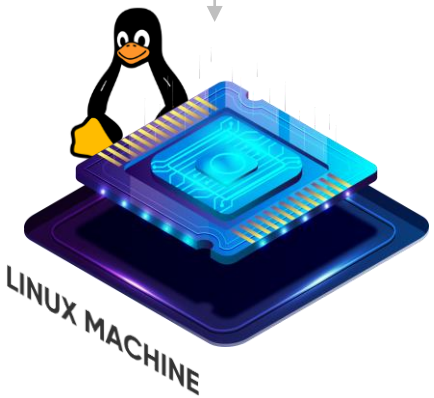
3. Request Signed Certificate

EZSSH

4. Return Signed Certificate

5. Return Signed Certificate

6. Access machine with Signed Certificate



KEYTOS

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

KEYTOS

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

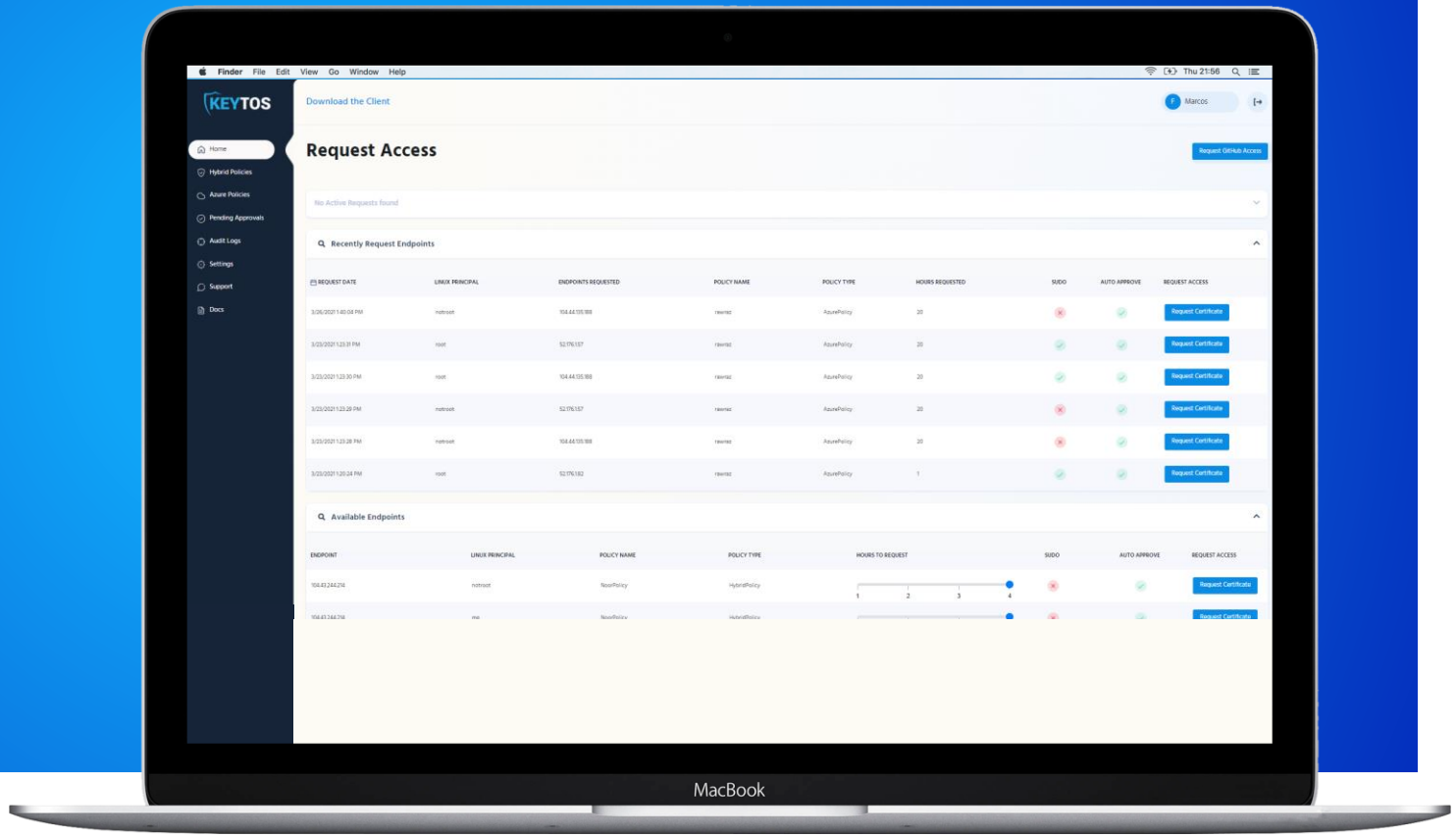
Reduce offboarding time and risk.

Native Linux Authentication no custom PAM module or code runs on your servers.

Bring your own Certificate Authority support

KEYTOS

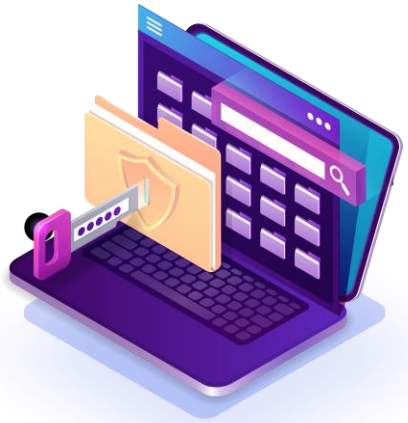
DEMO


KEYTOS
Download the Client
Request Access
Home
Hybrid Policies
Azure Policies
Pending Approvals
Audit Logs
Settings
Support
Docs
Recently Request Endpoints
Available Endpoints
MacBook

KEYTOS

HISTORY OF SSH AUTHENTICATION

1995 SSH Created



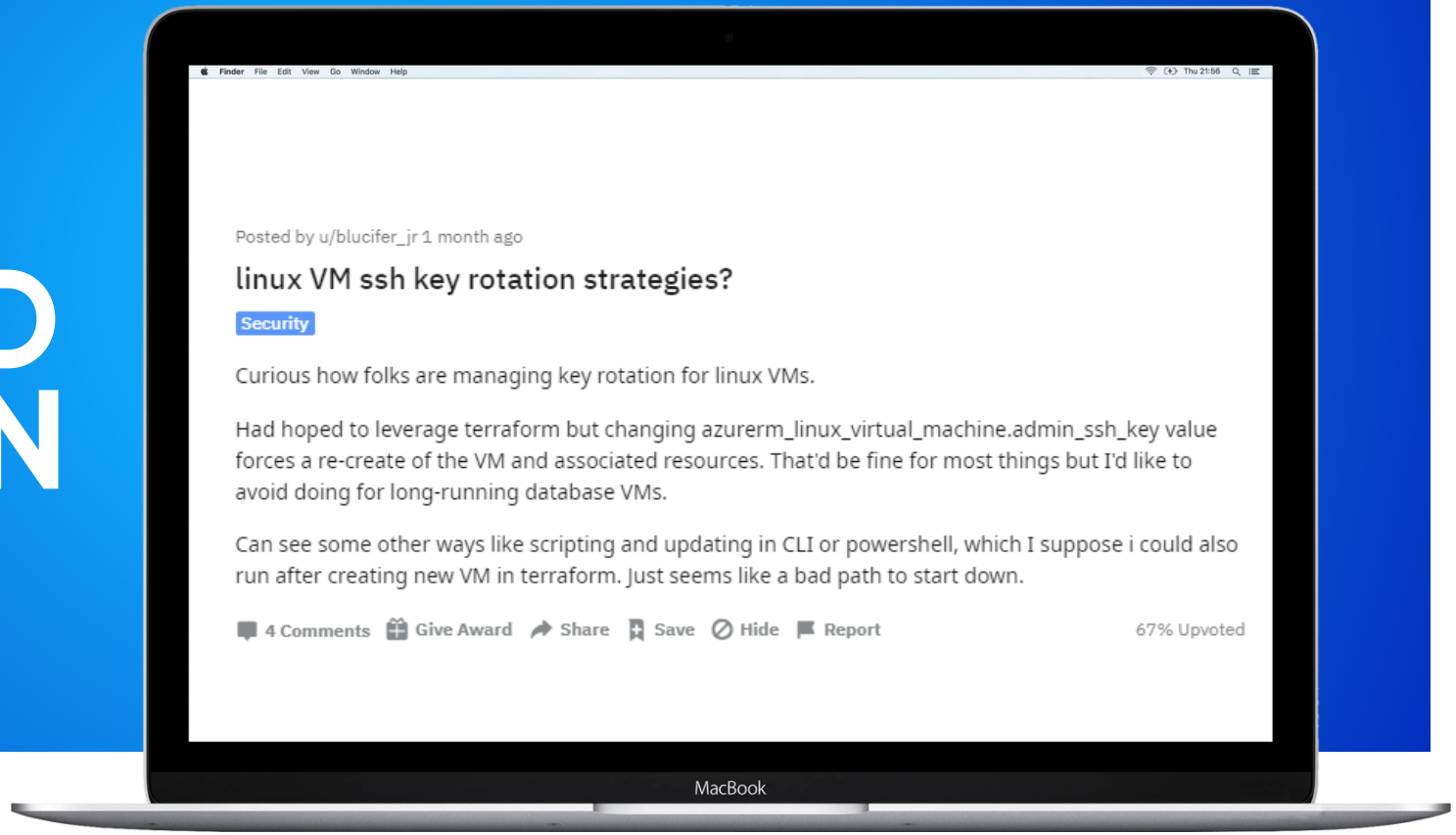
2010 SSH Certificates introduced

EZSSH

2021 EZSSH Makes SSH Certificates easy to use

KEYTOS

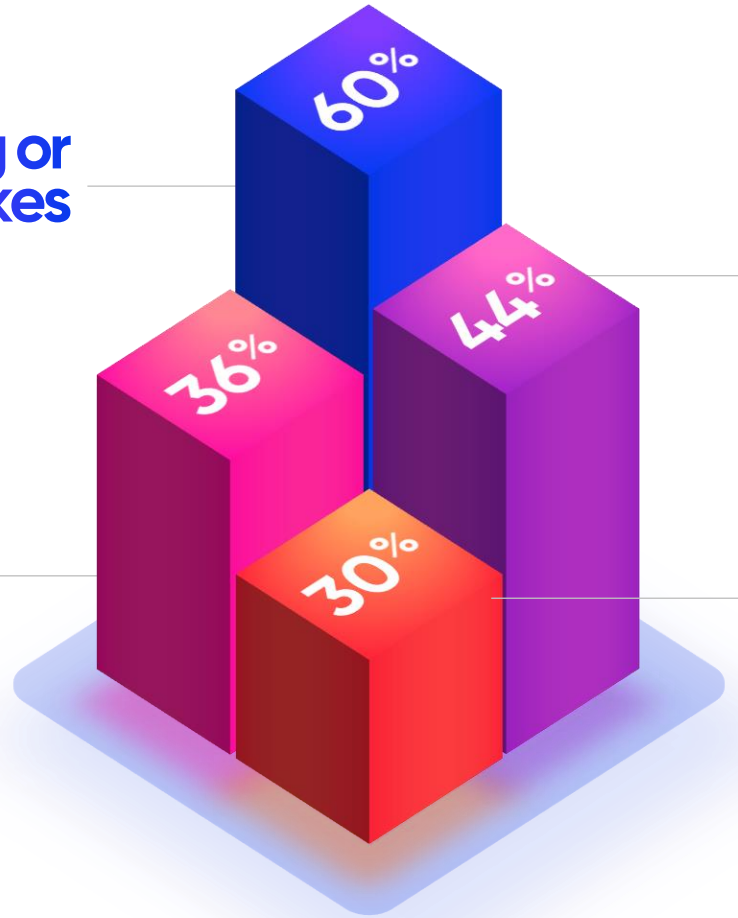
AZURE USERS NEED A SOLUTION



KEYTOS

ENGINEERS MAKE MISTAKES

The bulk of insider data breaches



Employees rushing or making mistakes

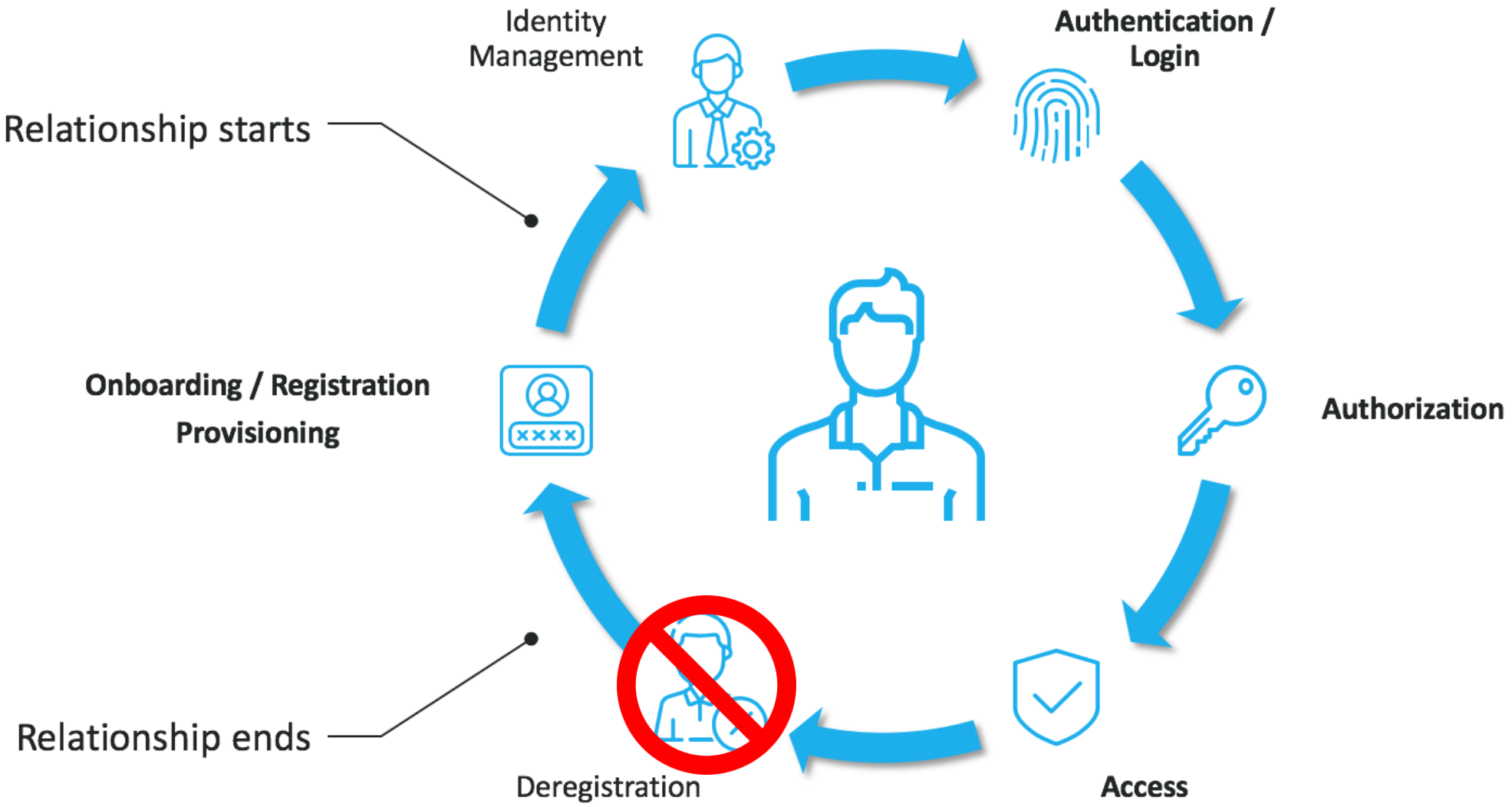
Lack of awareness

Lack of training

Intentionally leaking data

KEYTOS

IDENTITY LIFE CYCLE



KEYTOS

SSH UNDER ATTACK

- https://blog.ssh.com/ssh-key-scan-attack-honeypot
- https://www.zdnet.com/article/linux-under-attack-compromised-ssh-keys-lead-to-rootkit/
- https://securityaffairs.co/wordpress/37459/cyber-crime/compromised-ssh-keys.html
- https://www.beckershospitalreview.com/cybersecurity/github-leaks-exposed-up-to-200-000-medical-records-4-details.html

KEYTOS

EZSSH

Protecting GitHub with SSH Certificates

KEYTOS

PROBLEM OVERVIEW

- 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.

KEYTOS

GitHub Breaches

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

threatpost Cloud Security / Malware / Vulnerabilities / InfoSec Insiders / Po

Podcast: Shifting Cloud Security Left With Infrastructure-as-Code

Report: Microsoft's GitHub Account Gets Hacked

Home > News > Security > Source code from dozens of companies leaked online

Source code from dozens of companies leaked online

Compromised SSH keys used to access popular GitHub repositories

June 3, 2015 By Pierluigi Paganini

Security experts Ben Cox explained that the official Github repositories of the UK Government, Spotify, and Python were accessed using compromised SSH keys.

KEYTOS

OPPORTUNITY

- 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

Make audits easier with easy to audit logs

Reduce engineer onboarding time

Make security transparent for your users.

KEYTOS

OUR SOLUTION

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

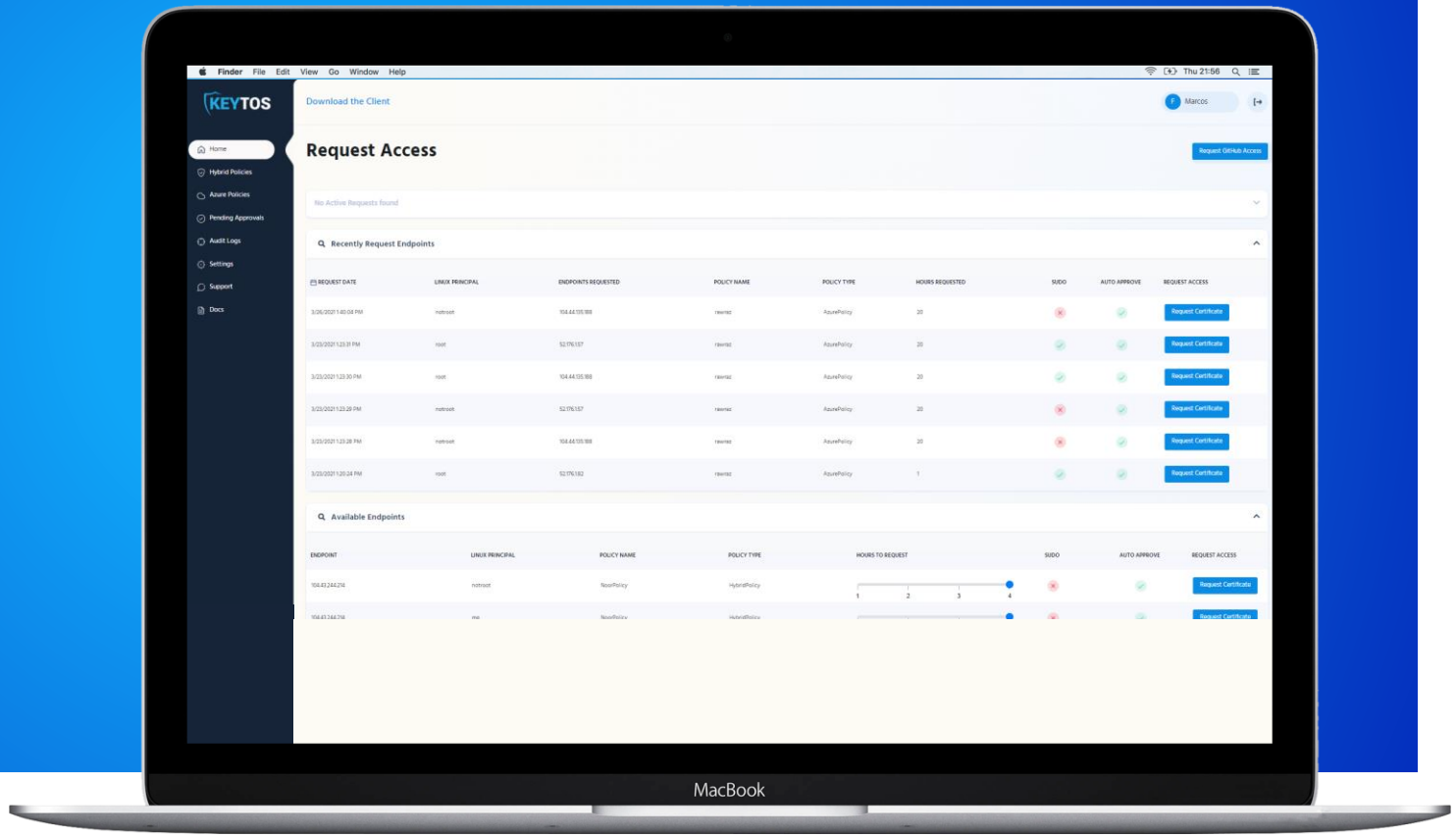
Automatically onboards new team members

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

Automatically removes access when no longer needed

KEYTOS

DEMO


KEYTOS
Download the Client
Request Access
Home
Hybrid Policies
Azure Policies
Pending Approvals
Audit Logs
Settings
Support
Docs
Recently Request Endpoints
Available Endpoints
MacBook

KEYTOS