ENGINEERING

Developer secrets need an extra layer of protection from code to cloud

Team: Engineering

Responsible for: Developing and building a company's products and/or internal and external applications

Shadow IT & Access Challenges:

- Developer secrets need an extra layer of protection because of privileged access to sensitive data and infrastructure
- Additional tools or services may be required to meet deadlines or work efficiently
- Perspective that security/IT policies and processes interfere with productivity and innovation
- Lack of visibility into the health and security of developer credentials like SSH keys

Your engineering team

is the heart of driving innovation and efficiency to build your company's product. They aim to deliver robust solutions that contribute to the organization's success and keep your company's solutions competitive. This team has access to some of your company's most critical infrastructure and data, and uses specialized secrets like SSH keys, API tokens, and other infrastructure secrets to access systems and integrate applications. It's critical to empower them to stay secure throughout the process.

Engineering and security

73% of developers agree that the work or tools their security team typically requires them to use interfere with their productivity and innovation[5]. In other words, if developers aren't able to handle sensitive company data in a streamlined, efficient manner, the organization is at risk.

73%

of developers feel that security tools interfere with productivity.

Software engineering teams are under immense pressure to deliver features on a strict schedule. Security policies or tools that slow down development directly impact productivity, so it's no surprise that developers don't want their day-to-day

productivity to be affected by security tools or requirements[6]. This is also why when developers are looking to increase productivity in their workflows, they may consider shortcuts (read: not follow security policy or embrace shadow IT).

Developers encounter distinctive challenges in their daily workflows, especially when it comes to the complexity involved with maintaining and securing developer secrets. For example, 60% of DevOps organizations have experienced secrets leakage in some form[7].

The key to securing the engineering team – and driving adoption of security tools and policies – is to reduce friction as much as possible. This goes for both enabling them to adopt the tools and services they need, as well as for securing secrets and access.

In many cases, there isn't a common practice for sharing sensitive values securely and efficiently. For example, developers may share secrets using unencrypted email and other messaging apps, manually set up system configurations on their local device to run a program, or manually copy sensitive values to connect to another machine.

Critically, offboarding must be considered when securing engineering teams. It's reported that 77% of developers still have access to a former employer's infrastructure secrets[8], leading to increased risk of breach. Code repositories hold company intellectual property, proprietary algorithms, or sensitive information, making it imperative that access is fully revoked in the offboarding process.