SecureCircle®

CASE STUDY

AWS Source Code Leak

Transparently Protect Source Code at All Times

AWS Source Code Leak

On January 13, an AWS employee stole nearly a gigabyte's worth of data, checking it into a personal GitHub repository[1]. This included data that had already been discovered and classified as "Amazon confidential". Even worse, the criminal was also able to steal RSA key pairs marked "admin" and "rootkey" (suggesting it provides privileged access control), passwords, auth tokens and API keys. Luckily for AWS, they were alerted to the theft by a 3rd party and took corrective action.

Source code breaches are occurring daily with damaging consequences. Uber and Google made headlines when a former Google engineer easily exfiltrated 9 GB of source code and hundreds of Waymo trade secrets. Tesla source code was breached by an employee, exfiltrating over 300,000 files for personal gain. Apple and Samsung suffered similar high-profile source code breaches.

Source code used to be something only tech geeks had to worry about, however today with the rise of the digital enterprise, source code is some of the most valuable corporate assets. Unfortunately, protecting source code in a way that doesn't impede development is one of the harder efforts in cyber security.

Source Code Protection requires Defense-in-Depth and Breadth

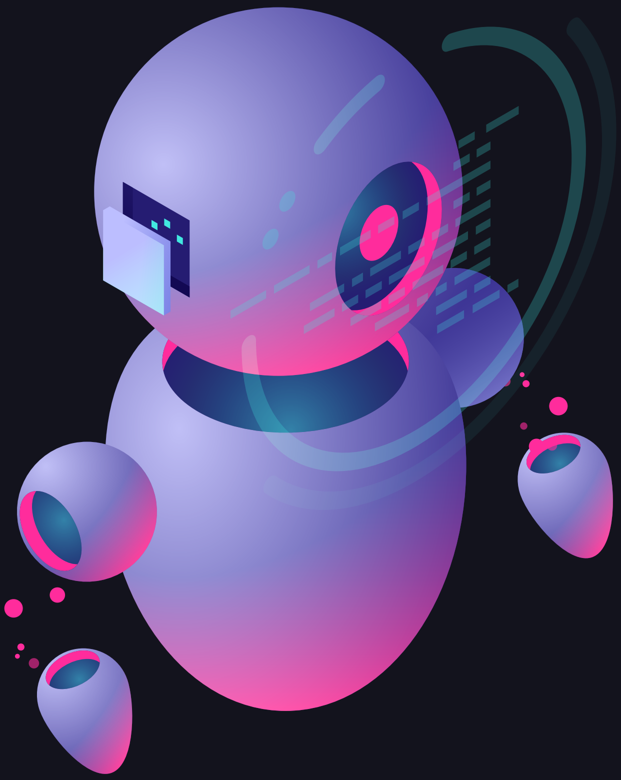
Protecting source code requires a defense-in-depth & breadth approach. The cybersecurity industry's traditional approach to data protection has focused on resolving the risks in depth. There are data loss prevention (DLP), information rights management (IRM), encryption and many other solutions available that can only protect a few pieces of data well. What organizations need is a security approach that enables a massive breadth of protection in addition to depth. It's critical to protect as much data as possible at all times. A breadth-of-security model has not been possible with legacy solutions because depth-of-security features interfere with developer tools and workflows.

Protecting the minimum amount of data, for the minimum amount of time, with the most restrictive workflows has hurt productivity and morale. Developers often find themselves fighting against security controls which slowdown their ability to accomplish the tasks they were hired to do. This leaves organizations particularly vulnerable to:

Source Code Protection requires Defense-in-Depth and Breadth

Human error.

This is the most common. Quite simply, data protection needs to make sure that developers can't accidentally check their source code into the wrong repository (a shockingly common problem given how most repositories work), or into an open source project. The AWS source code breach may also have been a form of human error, as a public statement by the company stated that the majority of the files leaked by the employee were personal in nature. Regardless of intent, a breach is still a breach.



Insider threat.

As in the Waymo and Tesla examples, a common threat vector can simply be a malicious employee or ex-employee who is motivated to breach the source code for personal gain. This can take several forms, including checking out the source code from the repository, and emailing it to a personal account, or moving it to a USB device.

Nation state attacker.

This happened recently to McAfee, Symantec and Trend Micro. Their source code was stolen by the Russian cybercriminal group Fxmsp.

Secure the Repository

For a solution to work at breadth, it needs to work at scale without limits. A complete defense-in-depth & breadth strategy ensures:

- All file types can be protected
- No file size limits
- Any application is supported
- Transparency to end users and no change to user workflows
- Portability across all devices and cloud
- Automatic tracking and protection of data as it moves
- Data is always protected, at-rest, in-transit and in-use
- Every derivative of the data is automatically protected

Developers Have Free Reign

Your source code is constantly being accessed by automated tool chains, internal developers, QA, 3rd party developers, and anyone or anything that has access to those associated systems. Even if all source code is stored in a repository with the appropriate data protection settings, your source code and copies of it are always in other locations. Even with properly configured repositories, the AWS, Waymo, and Tesla breaches occurred because source code must exist outside the repository. Without persistent data protection, once the data has been accessed, it's vulnerable to breach.

One of the reasons that source code protection is typically limited to the data at-rest, in the repository, is that development require specialized tools to create, build, and test code. Modern organizations have no tolerance for any additional workflow steps that will slow down productivity or force them to use anything, but the specialized tools required to get the job done. Business needs are fundamentally juxtaposed to the approach antiquated technologies, including DLP, DRM, and VDI, take. These approaches paralyze productivity with manual discovery and classification, limits on what tools can be used, and false positives that stall productivity.

Data Loss Prevention (DLP) requires a tremendous effort to build up and maintain rules which explicitly list what data to protect, when, how, etc. As a result, DLP is notorious for protecting only the bare minimum set of obvious data, such as predictably formatted credit card numbers, and even then, it is error-prone. The result is that DLP ends up focused on very basic and specific protection scenarios and is rarely able to protect source code. Which is just as well, as DLP imposes a heavy burden on employee productivity that engineers struggle to accept. We document the shortcomings of DLP at length in our whitepaper.

Digital rights management (DRM), such as Microsoft AIP, is not capable of protecting the majority of business data or something as complex as source code. DRM is too restrictive. DRM supports limited applications found in a normal business and is extremely limited in support for engineering environments.

Virtual Desktop Infrastructure (VDI) is quite secure. So secure that it's a burdensome solution for developers. VDI imposes too many restrictions and is very costly.

The result is that most companies simply do not implement data protection solutions to protect their source code. Those that do, introduce significant friction with developers, productivity is lost, and protection is highly error prone.

DASB: Transparently Protect Source Code At All Times

The Data Access Security Broker (DASB) platform provides extensive defense-in-depth and breadth for any data, even source code. When data leaves the repository, it is immediately and automatically encrypted. Data protected by DASB is transparently accessible to any IDE and development tools used. Protection is persistent no matter where the data goes or how it is accessed. The AWS, Waymo, and Tesla breaches would not have been possible with DASB. Moreover, once DASB is aware of some of your source code, it automatically protects any other similar data it comes in contact with, expansively extending DASB's protection automatically.

DASB operates transparently behind the scenes. Developers are not even aware that DASB is protecting the source code, unless they attempt to violate business policy. The organization has persistent access control over this valuable intellectual property and any derivatives which have been made. DASB tracks every action taken on protected data and reports it to your SIEM, turning every action into an auditable event.

Source code breaches are on the rise, and even the biggest players, like AWS, are not safe. AWS got lucky, but others like Waymo and Tesla suffered damaging losses. DASB would have prevented every one of these breaches, while providing a completely transparent experience to everyone involved. Don't get breached. Don't rely on luck. Rely on DASB.

Don't get breached.
Don't rely on luck.
Simply - rely on DASB.

[1] https://gizmodo.com/amazon-engineer-leaked-private-encryption-keys-outside-1841160934

About SecureCircle

SecureCircle's Data Access Security Broker (DASB) eliminates data breaches and mitigates insider threats, with no impact to the end-user experience and no modifications to applications and workflows. Data is always protected at rest, in-transit, and in-use; no matter where it is created, consumed, stored, modified, or shared. Headquartered in Silicon Valley, SecureCircle delivers the world's first data-centric protection for a zero-trust world.

SecureCircle®

SecureCircle.com
4701 Patrick Henry Drive
Building 19, Suite B,
Santa Clara, CA 95054
408-827-9100