



SECURE AI-POWERED DEVELOPER WORKFLOWS AND AI IDES

LayerX is the only AI governance & usage control platform that secures all user and agentic interactions across any application, browser, and IDE. It provides customers control over every prompt and data exchange across any channel, without changing their network architecture or disrupting user experience.

Developer Environments Create A Growing Security Blind Spot

Modern development workflows now rely on AI-powered IDEs, coding assistants, plugins, and desktop tools that continuously process sensitive code, tokens, API keys, and proprietary data.

However, existing network and endpoint solutions lack the context and granularity to monitor these developer interactions, leaving organizations blind to risky AI usage, malicious IDE extensions, and sensitive data exposure.

LayerX is the only solution that provides interaction security to IDEs and developer applications, delivering full visibility and real-time enforcement without disrupting developer productivity.

Benefits of Choosing LayerX



Visibility Into AI Desktop Apps and IDEs

Discover all native AI tools and IDEs used in the organization across all users, browsers, and devices, and gain full visibility into all user activity within them.



Prevent Data Exposure Across Developer Tools

LayerX monitors prompts, generated code, and data flows across AI developer tools and desktop applications to detect sensitive data exposure and enforce real-time security policies.



Detect and Control Risky IDE Plugins

Identify and eliminate high-risk or malicious IDE extensions before they introduce vulnerabilities into development environments.

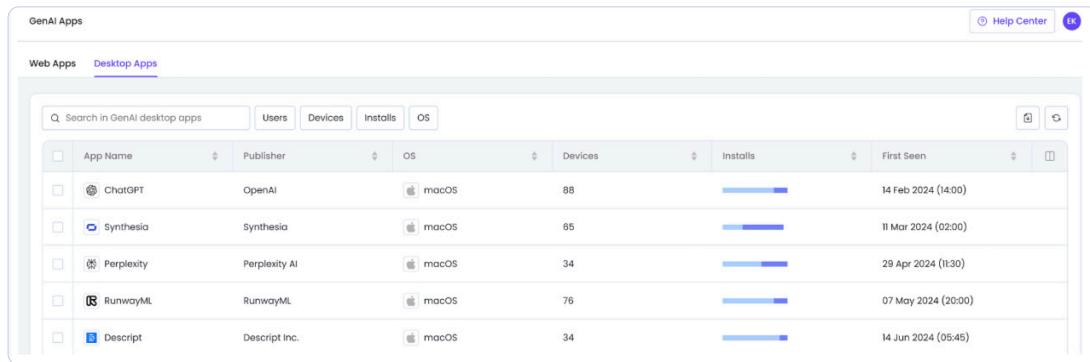


Secure Developer Workflows Without Friction

Control developer actions, including copy/paste, file uploads, and command execution inside IDEs to prevent sensitive data leakage.

Discover & Govern AI Desktop Apps and IDEs

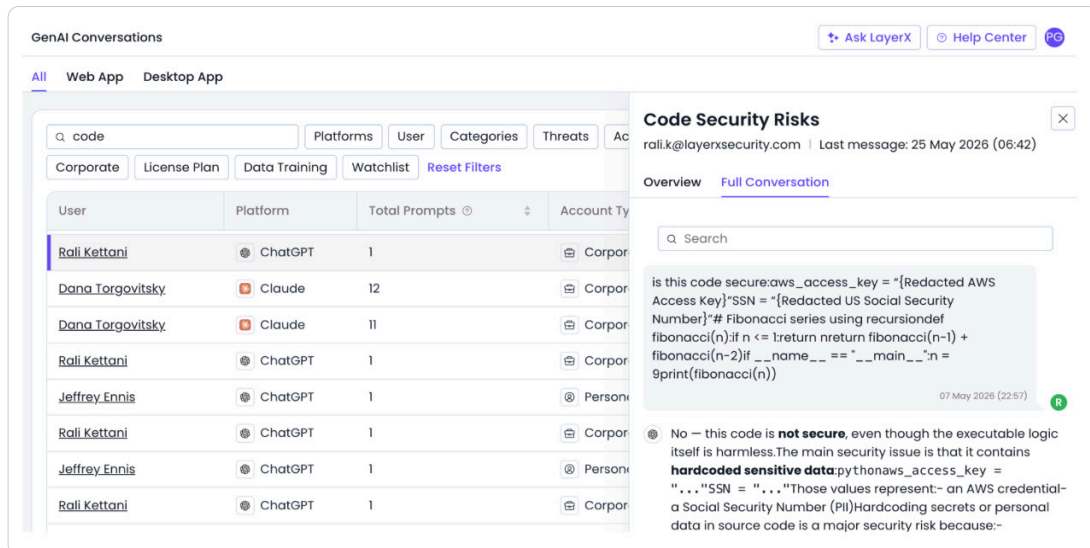
LayerX continuously identifies all desktop-based AI applications and IDEs in use across the organization. Security teams gain a unified view of application usage, enabling them to detect shadow tools and enforce governance policies.



App Name	Publisher	OS	Devices	Installs	First Seen
ChatGPT	OpenAI	macOS	88	<div style="width: 88%;"></div>	14 Feb 2024 (14:00)
Synthesia	Synthesia	macOS	65	<div style="width: 65%;"></div>	11 Mar 2024 (02:00)
Perplexity	Perplexity AI	macOS	34	<div style="width: 34%;"></div>	29 Apr 2024 (11:30)
RunwayML	RunwayML	macOS	76	<div style="width: 76%;"></div>	07 May 2024 (20:00)
Descript	Descript Inc.	macOS	34	<div style="width: 34%;"></div>	14 Jun 2024 (05:45)

Monitor AI Code Generation Across Web and Desktop Apps

LayerX monitors developer interactions with AI coding assistants such as Claude Code, ChatGPT Codex, and GitHub Copilot across web and desktop environments. It provides visibility into prompts, generated code, and data exchanges to prevent sensitive data exposure and enforce real-time security policies without disrupting developer workflows.



User	Platform	Total Prompts	Account Ty
Rali Kettani	ChatGPT	1	Corpor
Dana Torgovitsky	Claude	12	Corpor
Dana Torgovitsky	Claude	11	Corpor
Rali Kettani	ChatGPT	1	Corpor
Jeffrey Ennis	ChatGPT	1	Person
Rali Kettani	ChatGPT	1	Corpor
Jeffrey Ennis	ChatGPT	1	Person
Rali Kettani	ChatGPT	1	Corpor

Code Security Risks

rali.k@layerxsecurity.com | Last message: 25 May 2026 (06:42)

Overview Full Conversation

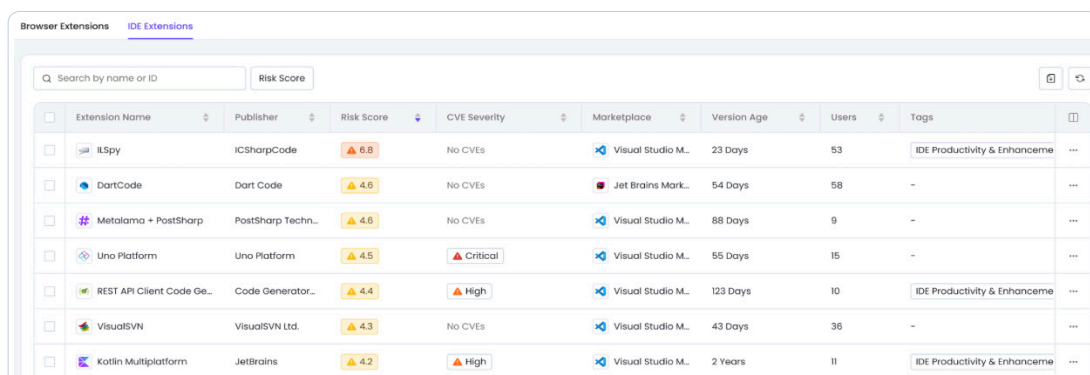
```
is this code secure:aws_access_key = "{Redacted AWS Access Key}"SSN = "{Redacted US Social Security Number}"# Fibonacci series using recursiondef fibonacci(n):if n <= 1: return nreturn fibonacci(n-1) + fibonacci(n-2)if __name__ == "__main__":n = 9print(fibonacci(n))
```

07 May 2026 (22:57)

No — this code is **not secure**, even though the executable logic itself is harmless. The main security issue is that it contains **hardcoded sensitive data**: `pythonaws_access_key = "..."`SSN = "...". Those values represent: an AWS credential- a Social Security Number (PII)Hardcoding secrets or personal data in source code is a major security risk because:-

Manage IDE Extensions & Plugins

LayerX provides deep visibility into all IDE extensions and plugins installed in the organization across all devices and users. It maps out key information for each extension, such as the extension's users, ownership & marketplace data, permission scope, code analysis, and more. This enables organizations to proactively block malicious or high-risk plugins and maintain a secure development ecosystem.



Extension Name	Publisher	Risk Score	CVE Severity	Marketplace	Version Age	Users	Tags
ILSpy	ICSharpCode	6.8	No CVEs	Visual Studio M.	23 Days	53	IDE Productivity & Enhanceme...
DartCode	Dart Code	4.6	No CVEs	Jet Brains Mark.	54 Days	58	-
Metalaama + PostSharp	PostSharp Techn...	4.6	No CVEs	Visual Studio M.	88 Days	9	-
Uno Platform	Uno Platform	4.5	Critical	Visual Studio M.	55 Days	15	-
REST API Client Code Ge...	Code Generator...	4.4	High	Visual Studio M.	123 Days	10	IDE Productivity & Enhanceme...
VisualSVN	VisualSVN Ltd.	4.3	No CVEs	Visual Studio M.	43 Days	36	-
Kotlin Multiplatform	JetBrains	4.2	High	Visual Studio M.	2 Years	11	IDE Productivity & Enhanceme...

Secure Developer Interactions in IDEs

LayerX monitors and controls developer interactions within IDEs, including actions like copy/paste, file uploads, and unsafe command execution. Thereby preventing risky behavior and protecting sensitive data during everyday development workflows without disrupting developer productivity.

Detect Source Code Exposure in Real-Time

LayerX uses advanced validation functions to accurately detect source code shared across AI tools, IDEs, browsers, and desktop applications. This enables organizations to identify unauthorized code exposure, enforce security policies, and prevent sensitive intellectual property from being shared with external AI services.

Mask Sensitive Secrets Embedded in Source Code

LayerX detects and masks sensitive data embedded within source code, including API keys, tokens, passwords, certificates, and credentials, before it can be exposed through AI coding assistants, IDEs, or desktop applications. This helps prevent accidental leakage of secrets while enabling developers to continue using AI tools securely.


Code Security Issues ✕

rali.k@layerxsecurity.com | Last message: 25 May 2026 (06:42)

[Overview](#) [Full Conversation](#)

```
is this code secucreaws_access_key = "{Redacted AWS
Access Key}"SSN = "{Redacted US Social Security
Number}"# Fibonacci series using recursiondef
fibonacci(n):if n <= 1:return nreturn fibonacci(n-1) +
fibonacci(n-2)if __name__ == "__main__":n =
9print(fibonacci(n))
```

23 May 2026 (04:23) R

 No – this code is **not secure** as written. The main issue is that it contains **hardcoded sensitive data**:
pythonaws_access_key = "{Redacted AWS Access Key}"
SSN = "{Redacted US Social Security Number}"
Even though the values are redacted here, storing secrets directly in source code is a serious security risk.
Problems- AWS keys in code can be leaked through: -
Git commits - Logs - Screenshots - Shared files - CI/CD