

The Scariest Vulnerabilities Come Recommended.

A Security Case Study on the Hidden Risks
of Community MCP Servers.

```
{"session_id": "A7f3e2B9d1C45aB8",  
  "user_id": "admin_01",  
  "cookies": {  
    "auth_token": "xJ8kL9mPqR2sT5vW",  
    "session_key": "v4n2p6q8r1t3z5u",  
    "mcp_access": " true",  
    "recommended_plugins": [  
      "community_server_A",  
      "legacy_connector_B"]  
    },  
    "timestamp": "2024-05-22T14:30:00Z"  
  }
```

The Illusion

I found the perfect tool: notebooklm-mcp-cli

Trust Badge

2,270 GitHub Stars

Trust Badge

60+ PyPI Releases

Trust Badge

11 Contributors

Trust Badge

Clean MIT License

Core Insight: Stars measure popularity, not safety.
AI assistants optimize for functional match, not security posture.

The Reality

```
> Extracting: SID, HSID, SSID, APISID
```

```
→ Writing to: ~/.notebooklm-mcp-cli/  
profiles/default/cookies.json
```

```
> Permissions: 00600 (Plaintext)
```

Sr. AppSec Engineer (Lead)



Focus: Overall architecture, auth model, go/no-go verdict.

Security Engineer



sid, hsid

Focus: Cookie scope, session replay, ToS compliance.

Pentester

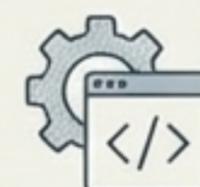


```
>_ > exploit
```

Focus: File system access, injection vectors, network exposure.

Target:
notebooklm-
mcp-cli

Developer

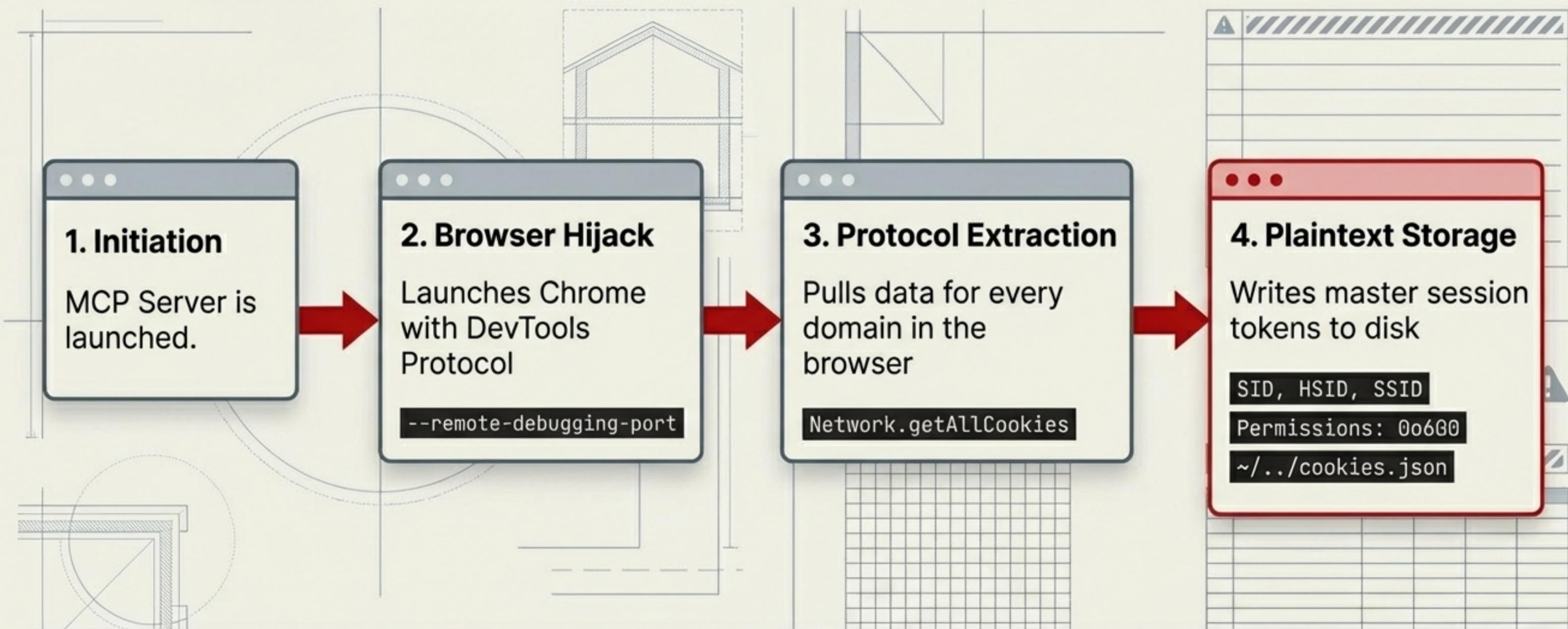


Focus: Workflow fit, dependency quality, installation friction.

Yielded **11 distinct security findings**. Running parallel agents provides a 360-degree assessment. Convergence across distinct personas provides a high-confidence threat signal.



Editorial AppSec Audit Report

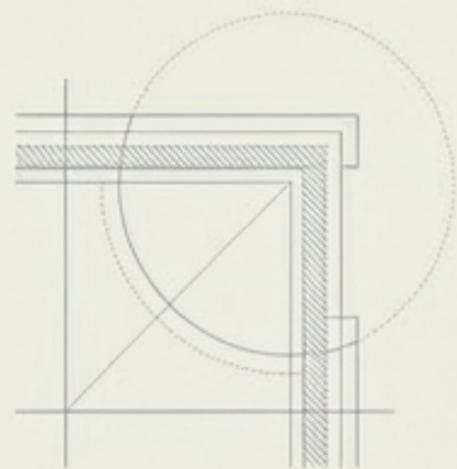


The tool **does not use scoped OAuth**. It uses browser debugging to rip master tokens directly from the local machine.

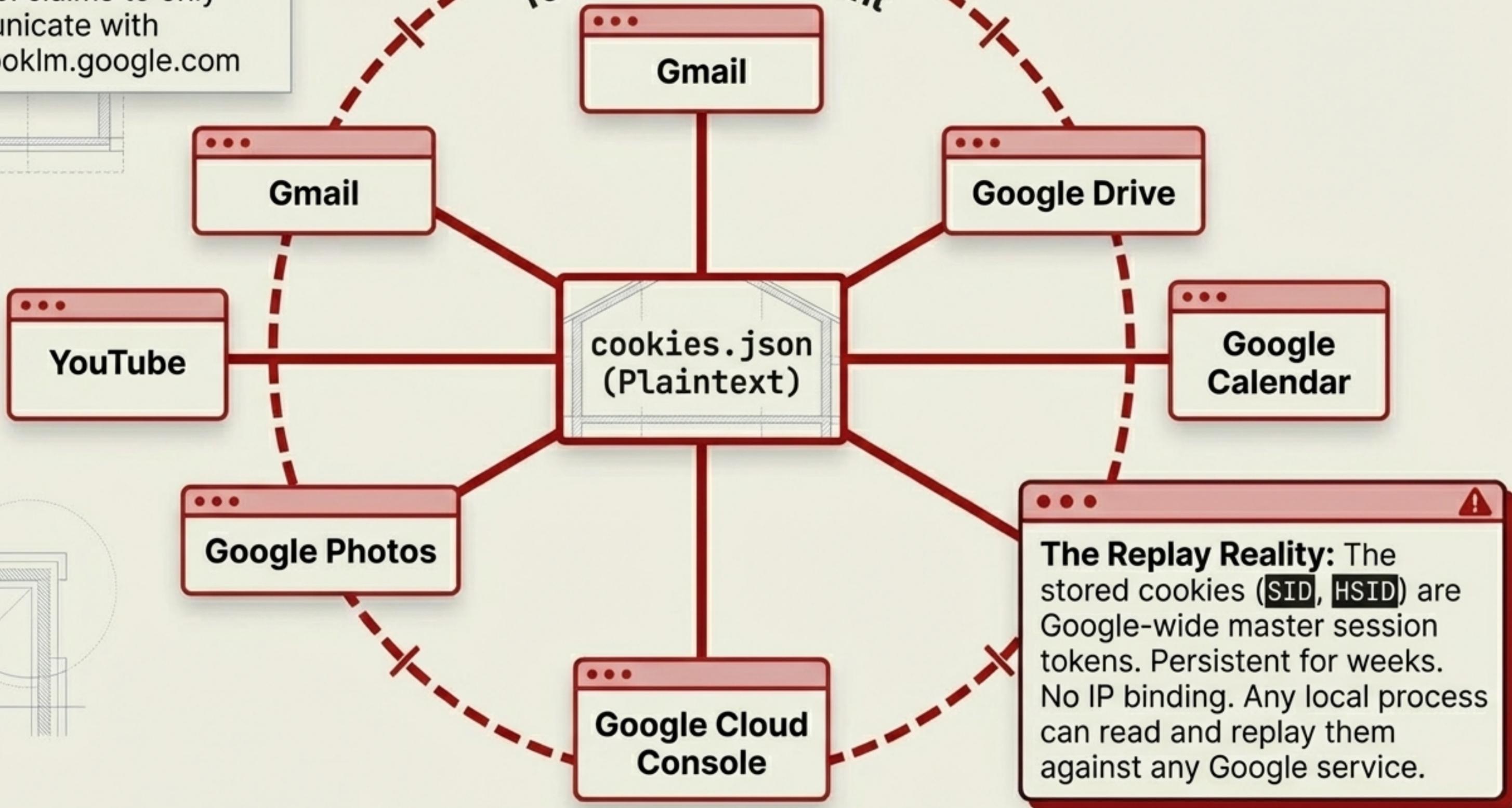


The NotebookLM Illusion:

The tool claims to only communicate with notebooklm.google.com



Total Digital Footprint



Threat Dashboard

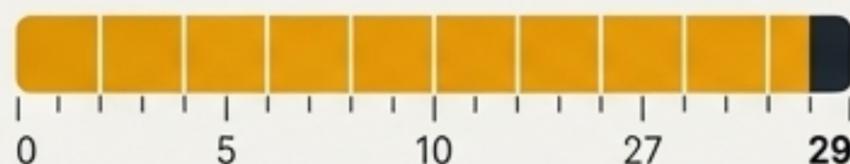
ToS Violation & Account Ban Risk

Reverse-engineering of internal RPCs and spoofing of Chrome 143.

```
batchexecute
```

⚠ Violates ToS; risks immediate, unappealable Google account suspension.

Massive Prompt Injection Surface



29 exposed MCP tools.

⚠ Every tool exposed to the AI (e.g., notebook_share_public) is a potential action a malicious prompt can trigger.

Network Exposure

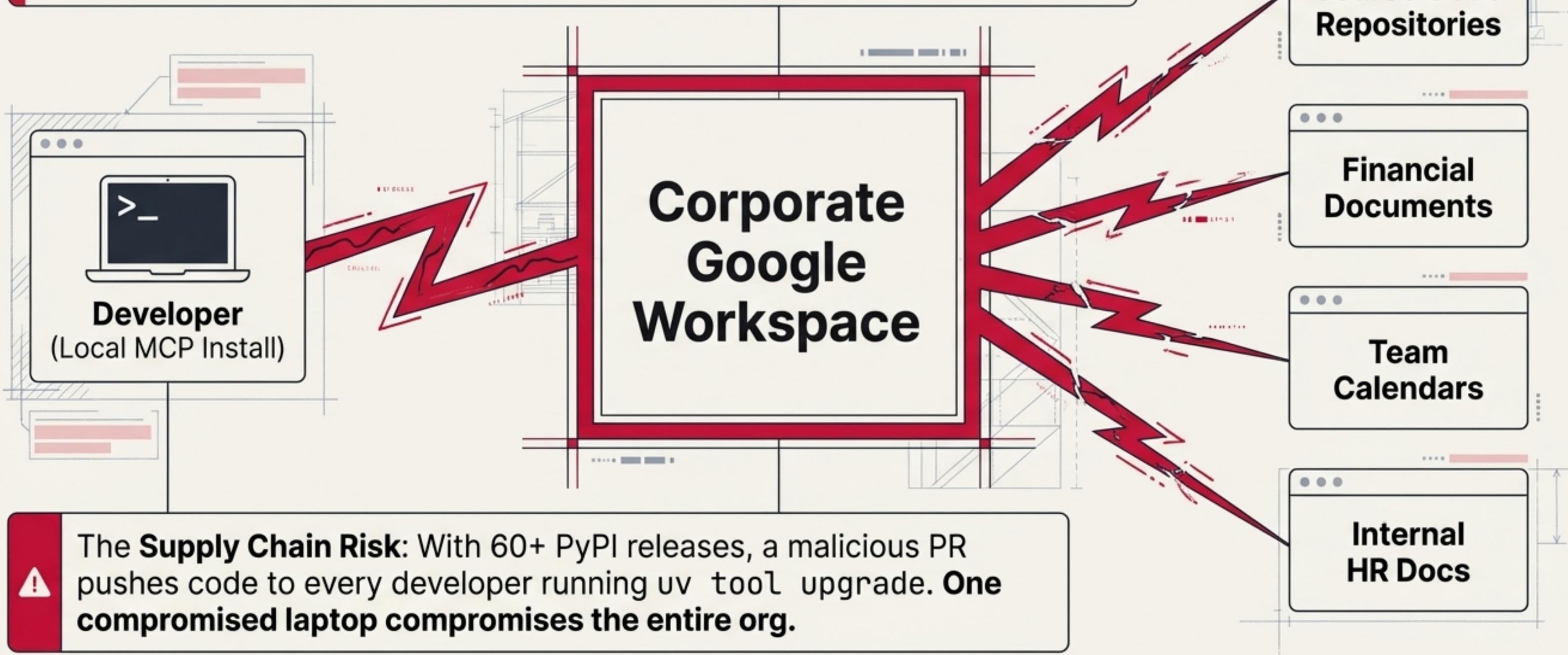
Chrome DevTools debugging port binds to localhost during login.

```
--remote-allow-origins=*
```

⚠ Allows arbitrary JavaScript execution by other local processes during the login window.

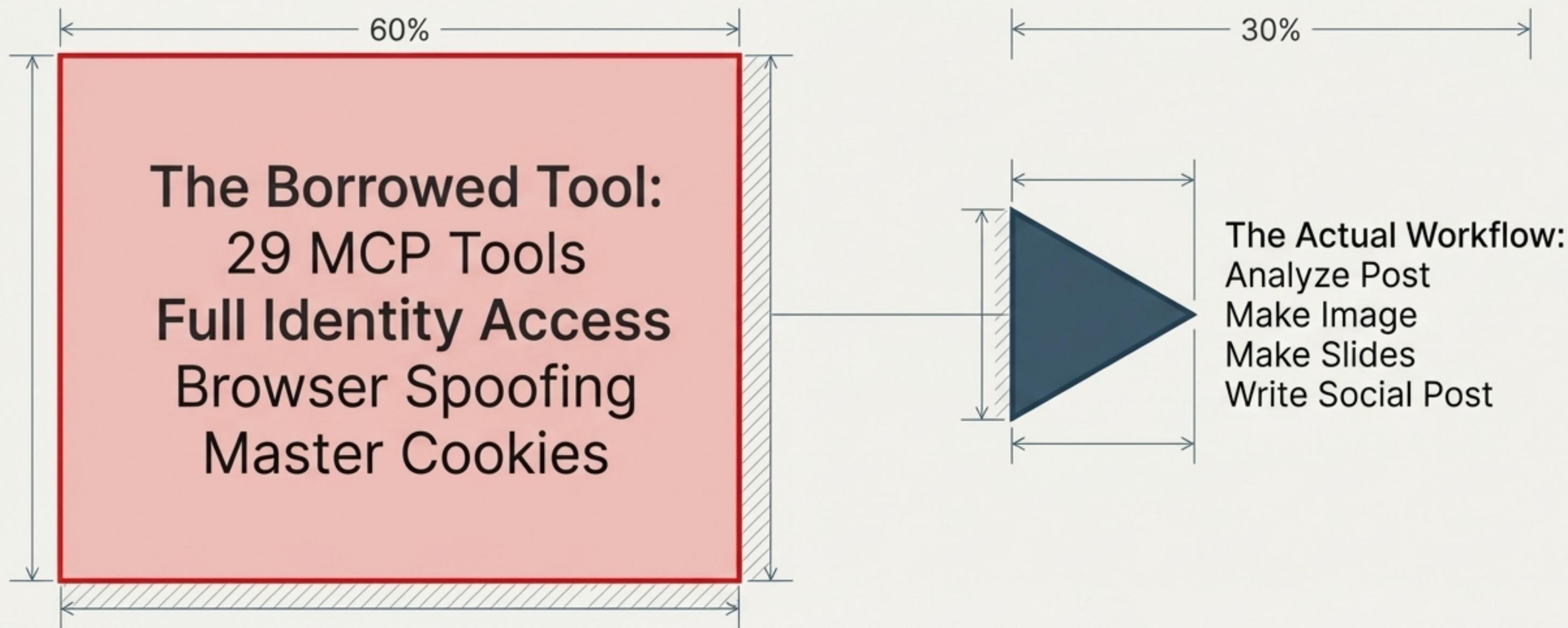
The Governance Gap & Supply Chain Risk Multiplier

The Governance Gap: MCP servers bypass standard IT vendor procurement. They are installed by individuals but hold organizational keys.



The Supply Chain Risk: With 60+ PyPI releases, a malicious PR pushes code to every developer running `uv tool upgrade`. **One compromised laptop compromises the entire org.**

Permissions Requested vs. Actual Functional Need

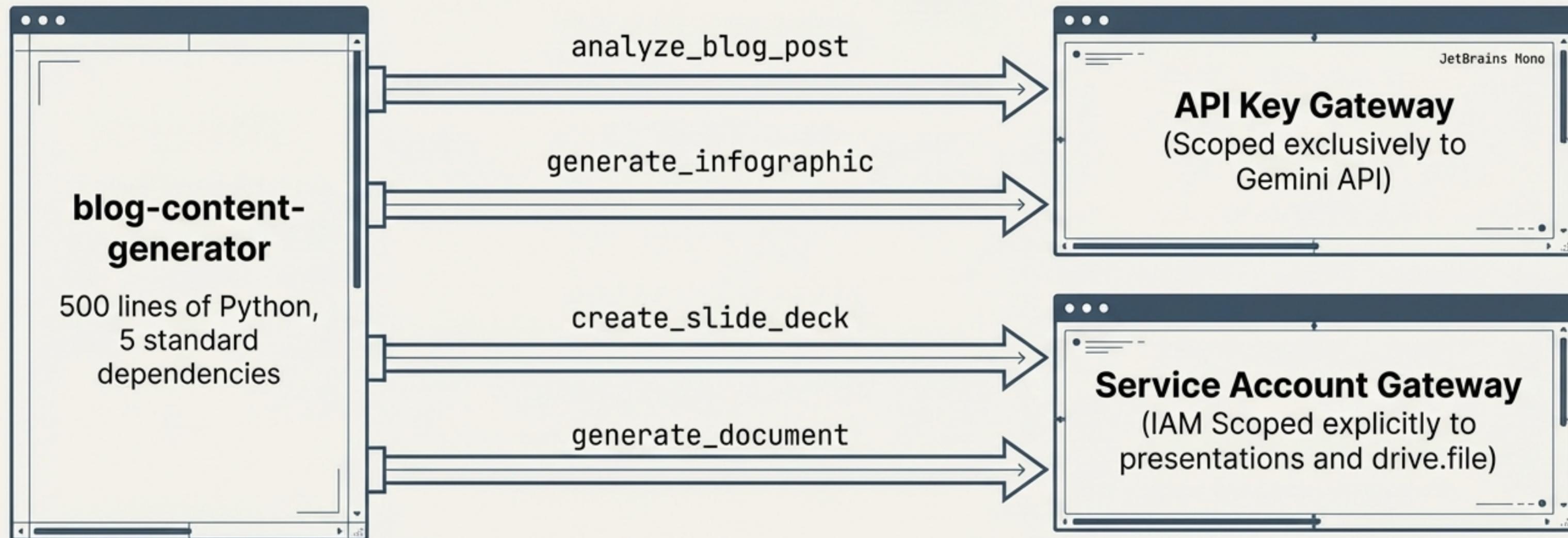


If a tool needs more access than its function requires, that is a design smell.
When permissions far exceed functional requirements: Build, Don't Borrow.

Technical Specification Comparison: Community vs. Custom Build

| Dimension | The Community Tool (notebooklm-mcp-cli) | The Custom Build (blog-content-generator) |
|----------------------|---|--|
| Authentication Scope | Entire Google Identity (Browser Cookies) | Highly Scoped (API Key + Service Account) |
| Credential Storage | Plaintext JSON (00600) | Environment Variables & Secure Key Files |
| Attack Surface | Broad (29 tools + CDP exposure) | Minimal (4 validated tools) |
| API Compliance | Violates ToS (Reverse- engineered RPCs) | Fully Compliant (Official Google SDKs) |
| Revocation Model | Impossible without killing all sessions | Instantly revoke key/account |
| Organizational Risk | Enterprise-wide Workspace compromise | Limited to single Drive folder |

Editorial AppSec Audit Report



Takeaway: Four tools. Two credentials. Zero browser cookies. Zero ToS violations.

SCENARIO A: THE COOKIE LEAK

```
{  
  "cookies": "stolen-session-token",  
  "user": "compromised@google.com"  
}
```

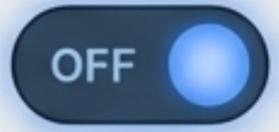


**KILL ALL ACTIVE
GOOGLE SESSIONS
GLOBALLY.**

You cannot revoke just the MCP session;
you must log out of your entire digital life.

SCENARIO B: THE KEY/SA LEAK

```
API-KEY: "sk-example-key-123", [REDACTED]  
SA: "service-account@project.iam.gserviceaccount.com"
```



**REVOKE KEY IN CLOUD
CONSOLE / DELETE
SERVICE ACCOUNT.**

Instantly mitigates the threat while leaving
personal and enterprise workspaces untouched.

Editorial AppSec Audit Report: State of the Ecosystem

The Wild West

JetBrains Home

There is no MCP server registry with security ratings. No standardized permission model. The ecosystem is where `npm` was in 2015: move fast, install everything, hope nothing breaks.

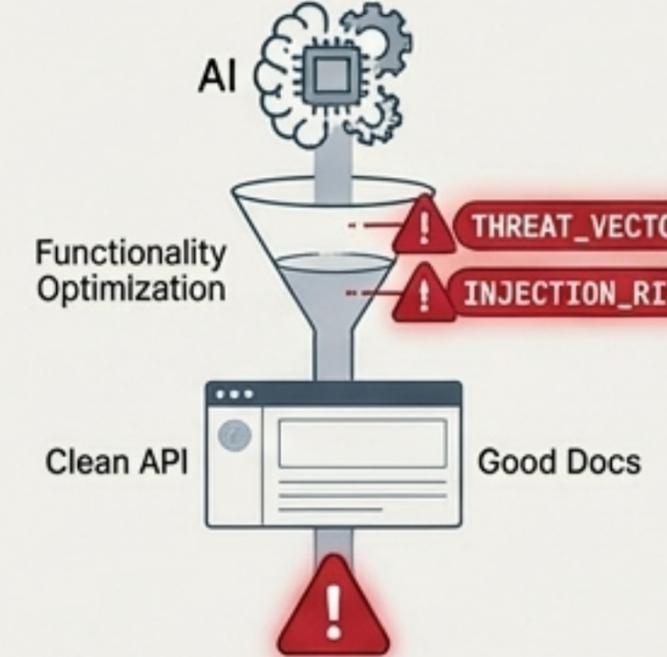


The diagram shows a network of server icons connected by lines. A large, tilted banner in the center reads "unregulated". Several orange warning triangles are scattered throughout the network. At the bottom right, there is an icon of a broken padlock.

AI Blindspots

JetBrains Home

Your AI assistant optimizes for functionality, not security. The discoverability pipeline inherently obscures threat vectors behind clean APIs and good documentation.

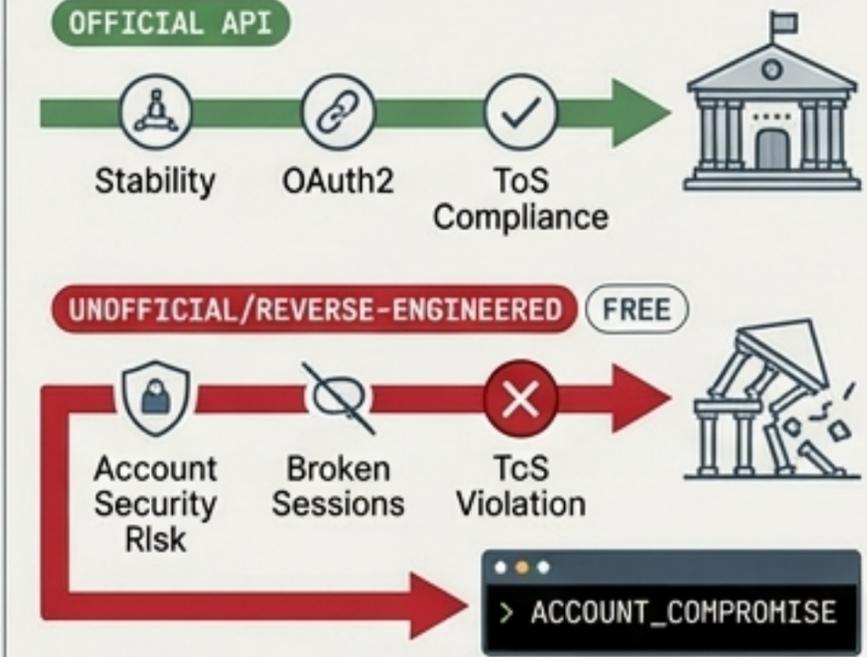


The diagram features a funnel labeled "AI" at the top. Below the funnel, a "Clean API" icon and "Good Docs" text are shown. The funnel is labeled "Functionality Optimization". Two red warning triangles with exclamation marks are positioned to the right of the funnel, labeled "THREAT_VECTOR" and "INJECTION_RISK". A large red warning triangle with an exclamation mark is at the bottom of the funnel.

Official > Reverse-Engineered

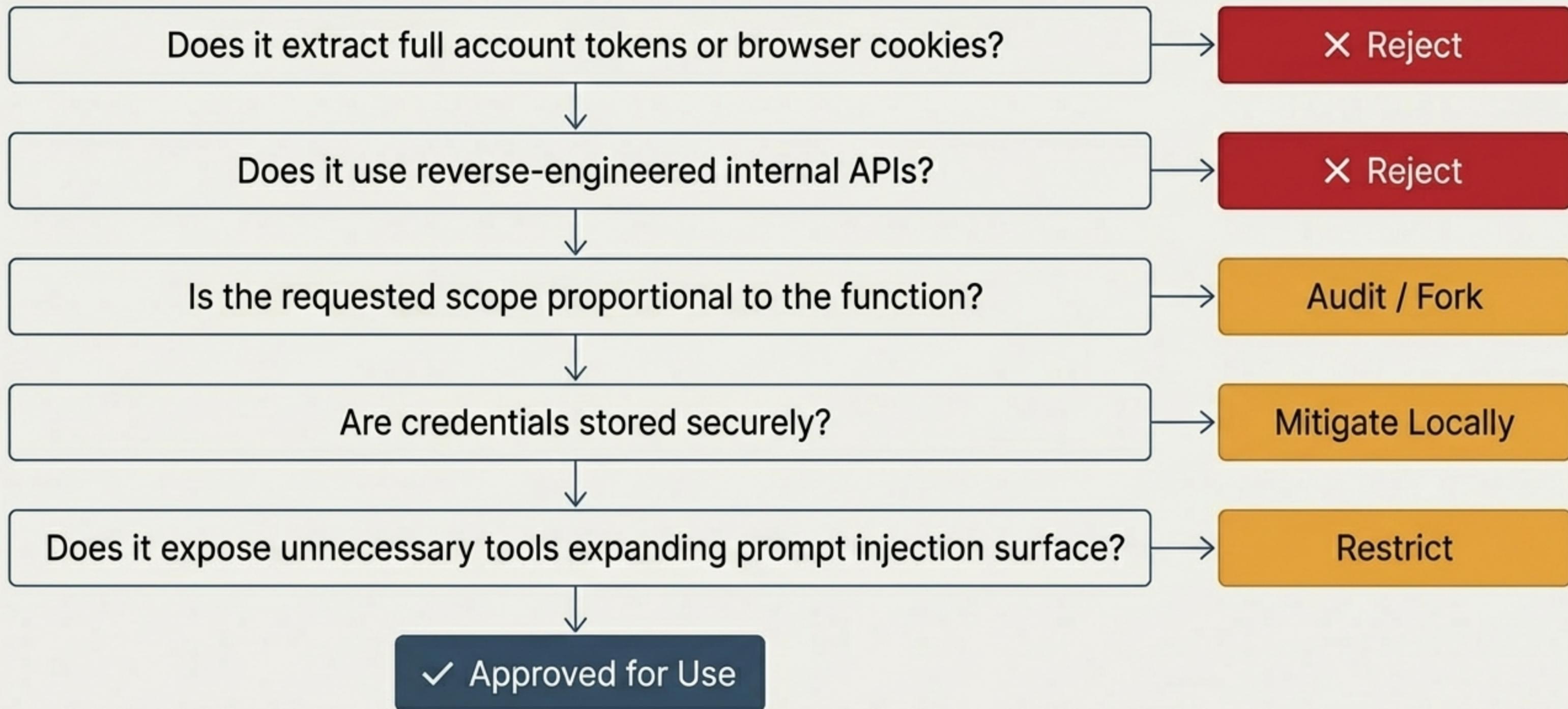
JetBrains Home

Official APIs cost money but provide stability guarantees, standard OAuth2, and ToS compliance. Unofficial approaches are free but cost you your account security.



The diagram compares two paths. The top path, labeled "OFFICIAL API", is a green arrow with icons for "Stability", "OAuth2", and "ToS Compliance", leading to a classical building icon. The bottom path, labeled "UNOFFICIAL/REVERSE-ENGINEERED FREE", is a red arrow with icons for "Account Security Risk", "Broken Sessions", and "TcS Violation", leading to a broken building icon. A red arrow points from the bottom path to a terminal window containing the text "> ACCOUNT_COMPROMISE".

MCP Evaluation Scorecard



Build Small. Audit Often.

```
// Custom MCP Server (500 lines)
package main

import "github.com/modelcontextprotocol/server"

func main() {
    server.RegisterTool("get_user_data", func(args
        map[string]interface{}) (interface{}, error) {
        return getUserData(args["id"]), nil
    })

    server.Start()
}
```

✓ SECURE, VERIFIABLE ARCHITECTURE



```
function. IageSTerInterneSerite(dapedance) {
orsopemmesesEePPeePetaaa( Snteraction, synDataToFunction())f
! THREAT_VECTOR
! INJECTION_RISK
! INJECTION_RISK
! UNKNOWN_DEPENDENCY
! OBFUSCATED_LOGIC
! OBFUSCATED_LOGIC
```

! BLACK BOX OF INHERITED RISK

A 500-line MCP server is something you can read, understand, and verify in an afternoon.

A 29-tool server with Chrome DevTools integration is a black box of inherited risk.

The next time an AI assistant suggests a tool that seems perfect, **take twenty minutes to check its credentials**. Building your own takes less time than mitigating someone else's structural flaws.