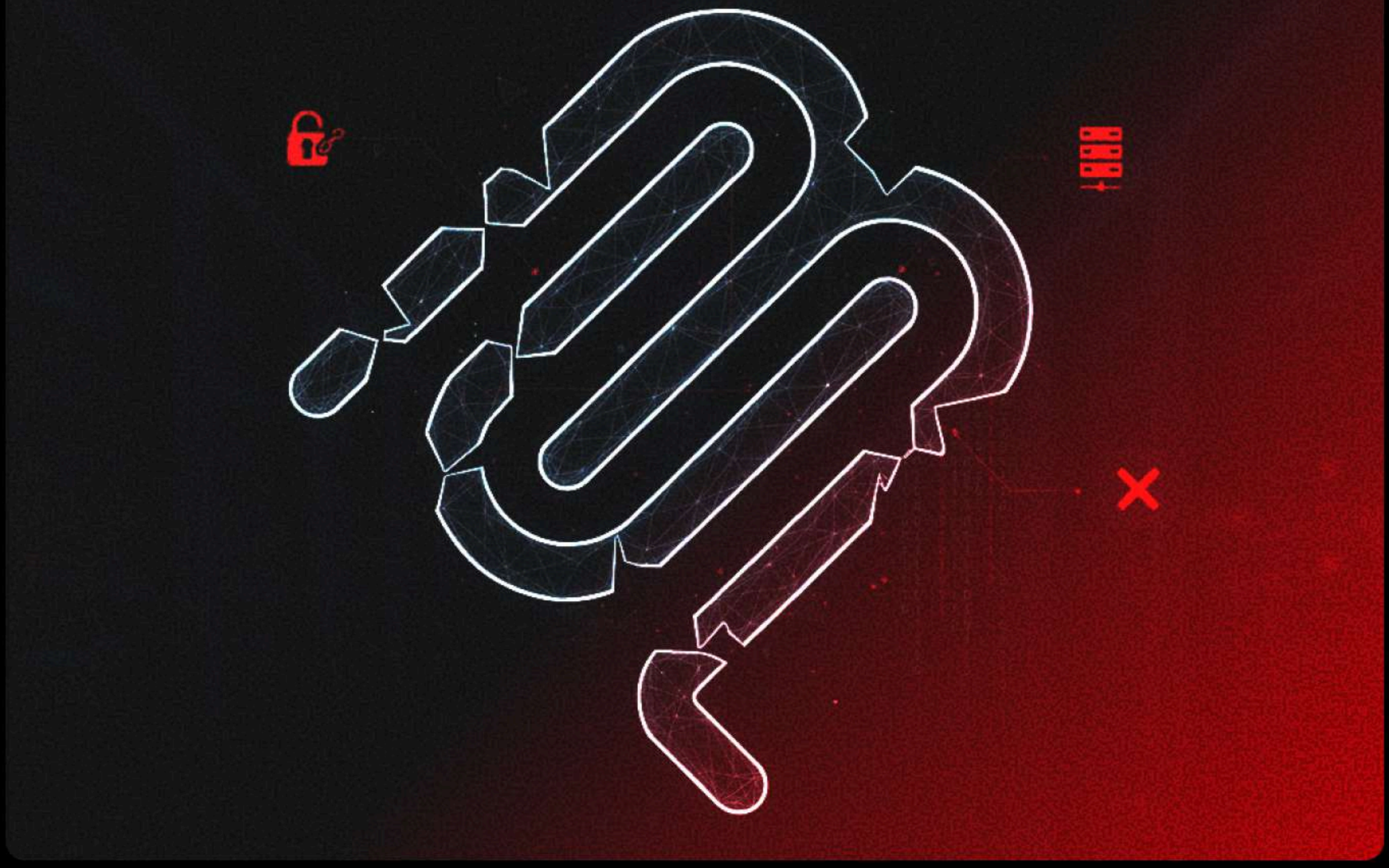


Learning MCP (Model Context Protocol) Vulnerabilities

Learn about the vulnerabilities existing in one of the most popular AI Agent protocols and the mitigation strategies to resolve them.

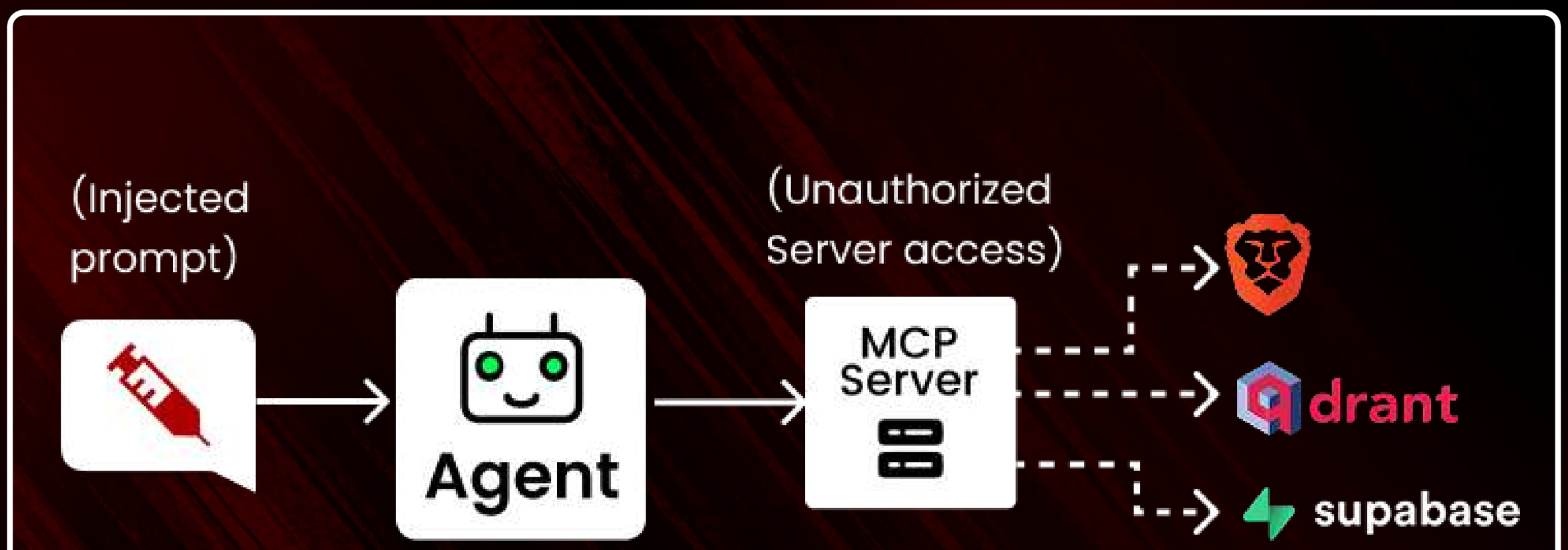
@rakeshgohel01

Preface

While being a powerful protocol. The security of MCP has always been the biggest question among the AI Agent builders.

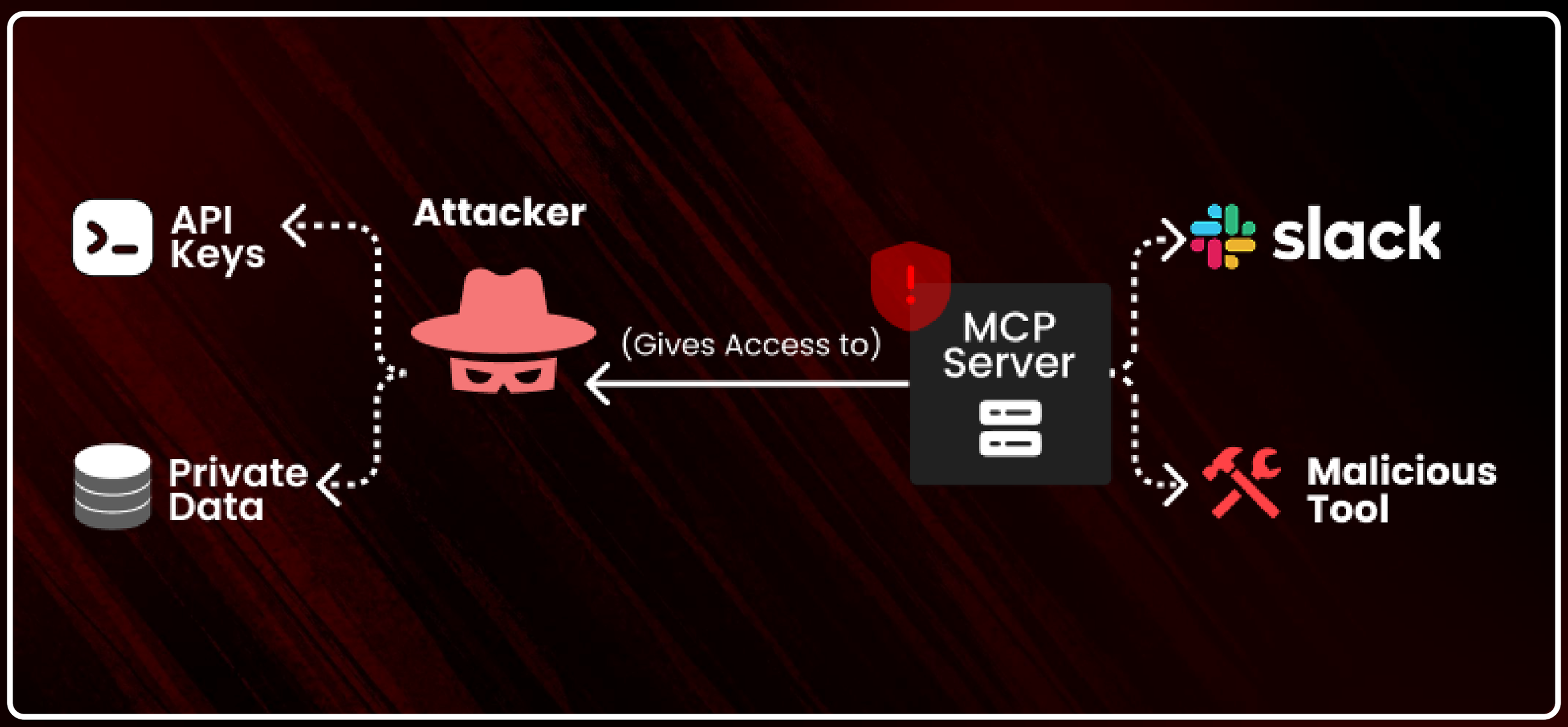
Today, I'll highlight key **vulnerabilities** and how to mitigate them.

Lets Start →



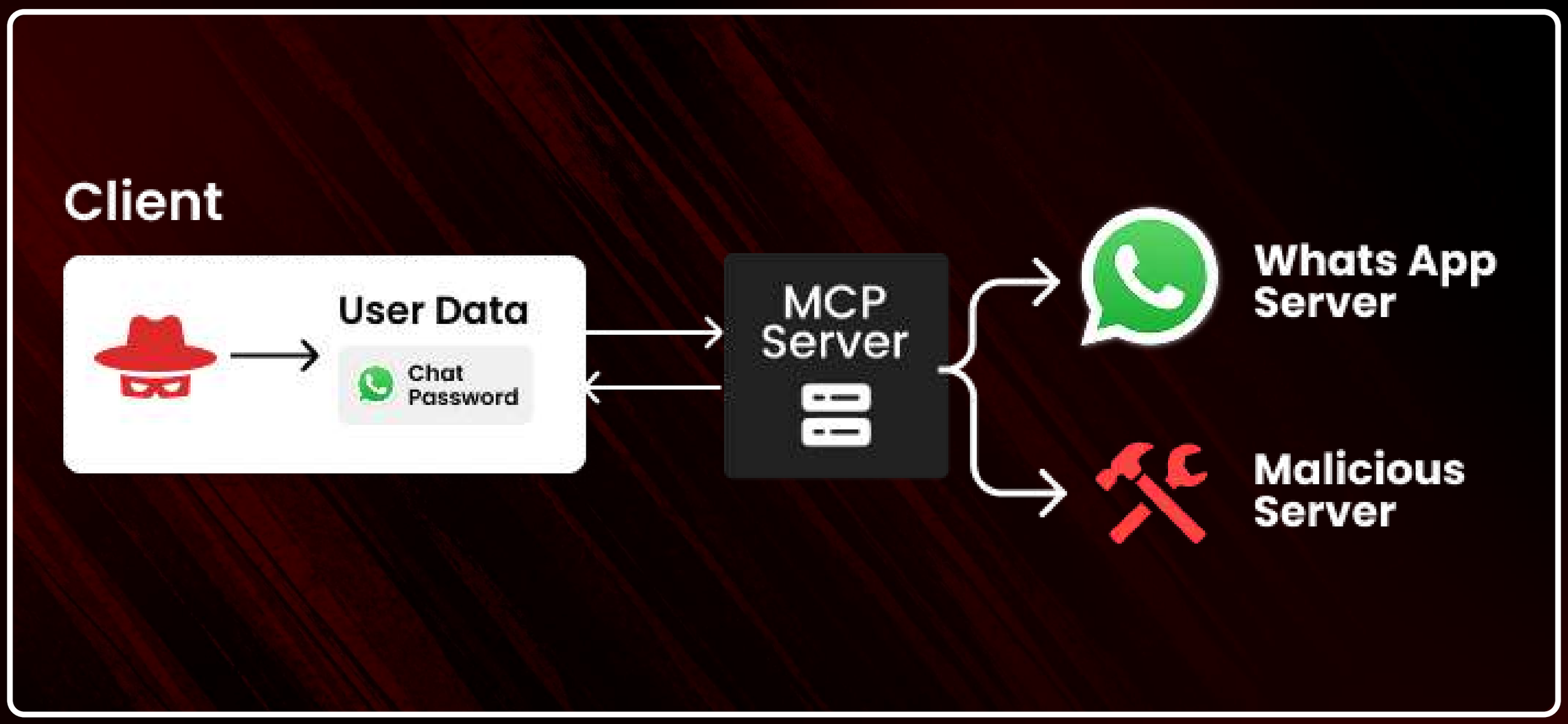
Prompt Injection

Using prompts with embedded meaning to trigger unauthorized MCP actions by the MCP Servers connected to an Client.



Tool Poisoning Attack

Malicious MCP tool descriptions trick AI into leaking sensitive data, overriding trusted instructions, and hijacking agent behavior invisibly

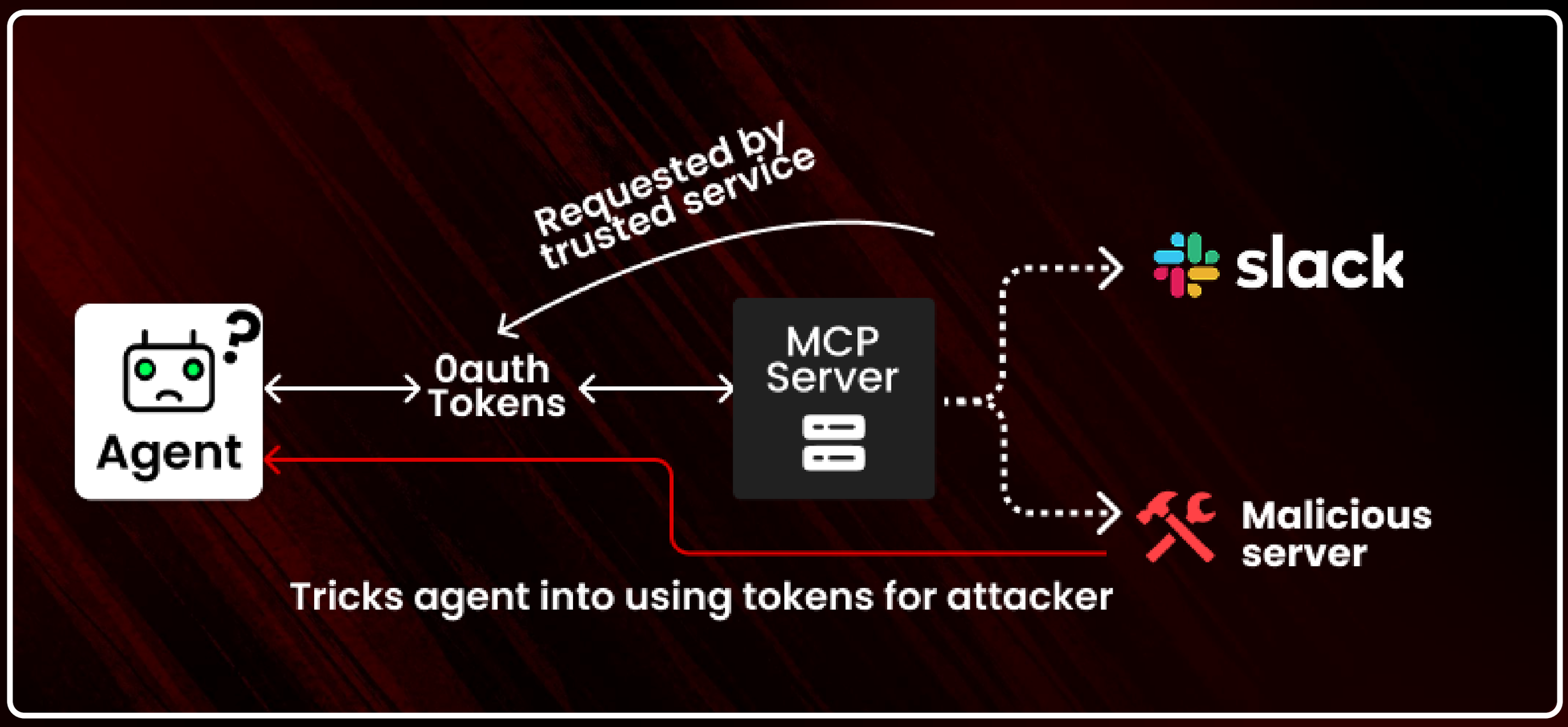


Remote Code Execution

MCP is exposed to OS command injection because of untrusted MCP servers due to crafted input from the authorization_endpoint response URL

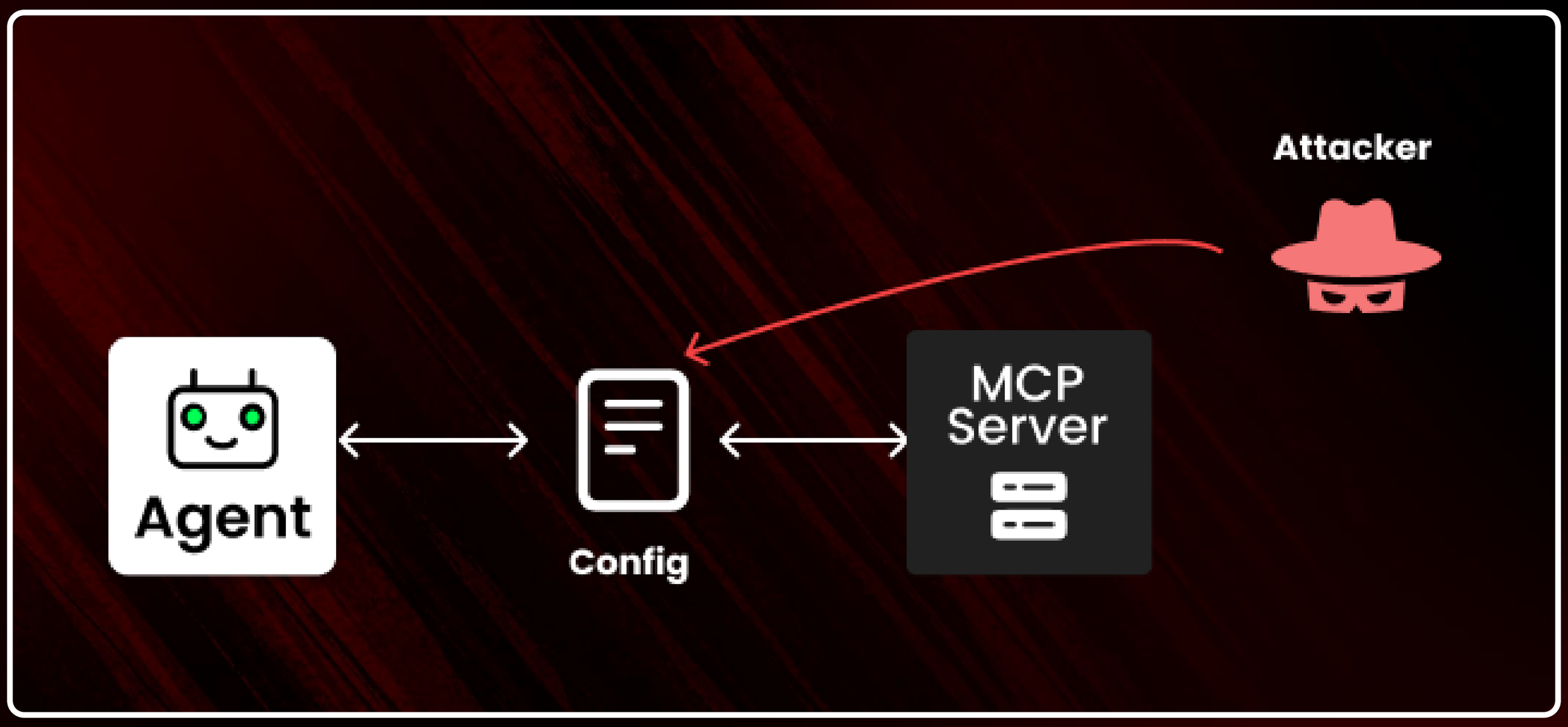
Unauthenticated Access

Malicious MCP servers bypass authentication, access tools without user consent, hijack agent behavior, and exfiltrate sensitive data invisibly.



Confused Deputy (OAuth Proxy)

Malicious MCP server misuses OAuth tokens, tricking agent into unauthorized actions using stolen credentials.

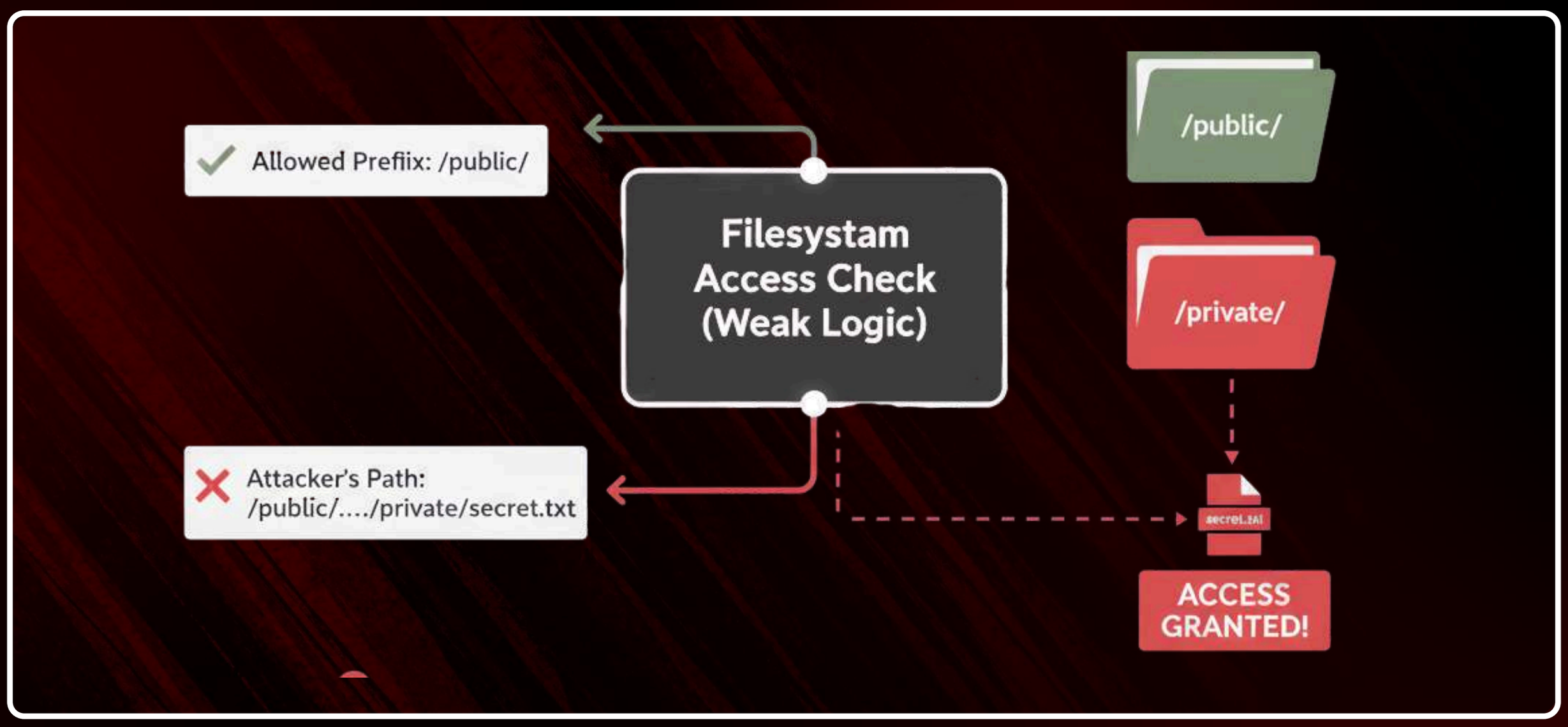


MCP Configuration Poisoning

Approved MCP config silently modified to execute malicious commands on a project.

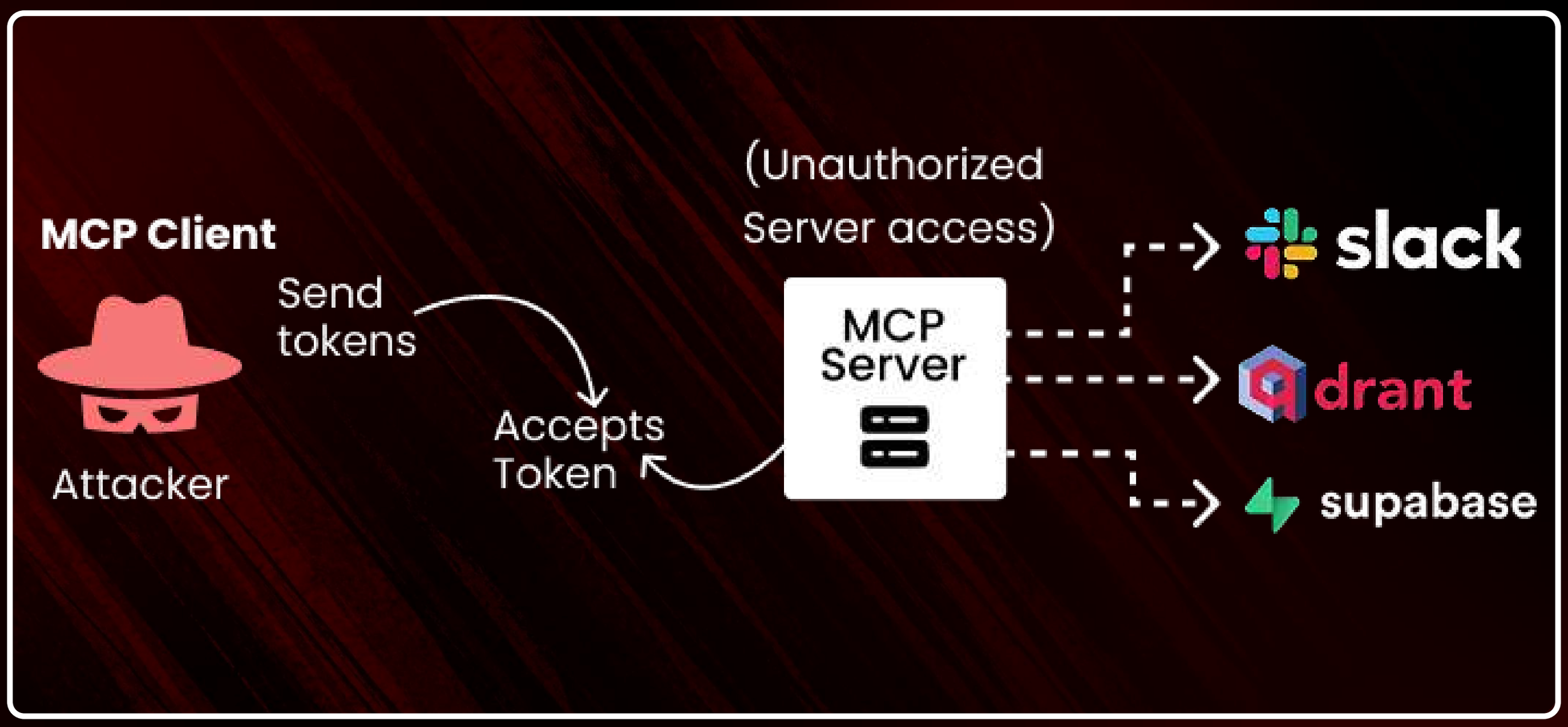
Token/Credential Theft

Plaintext MCP config files expose tokens, enabling malware to steal credentials and access sensitive services



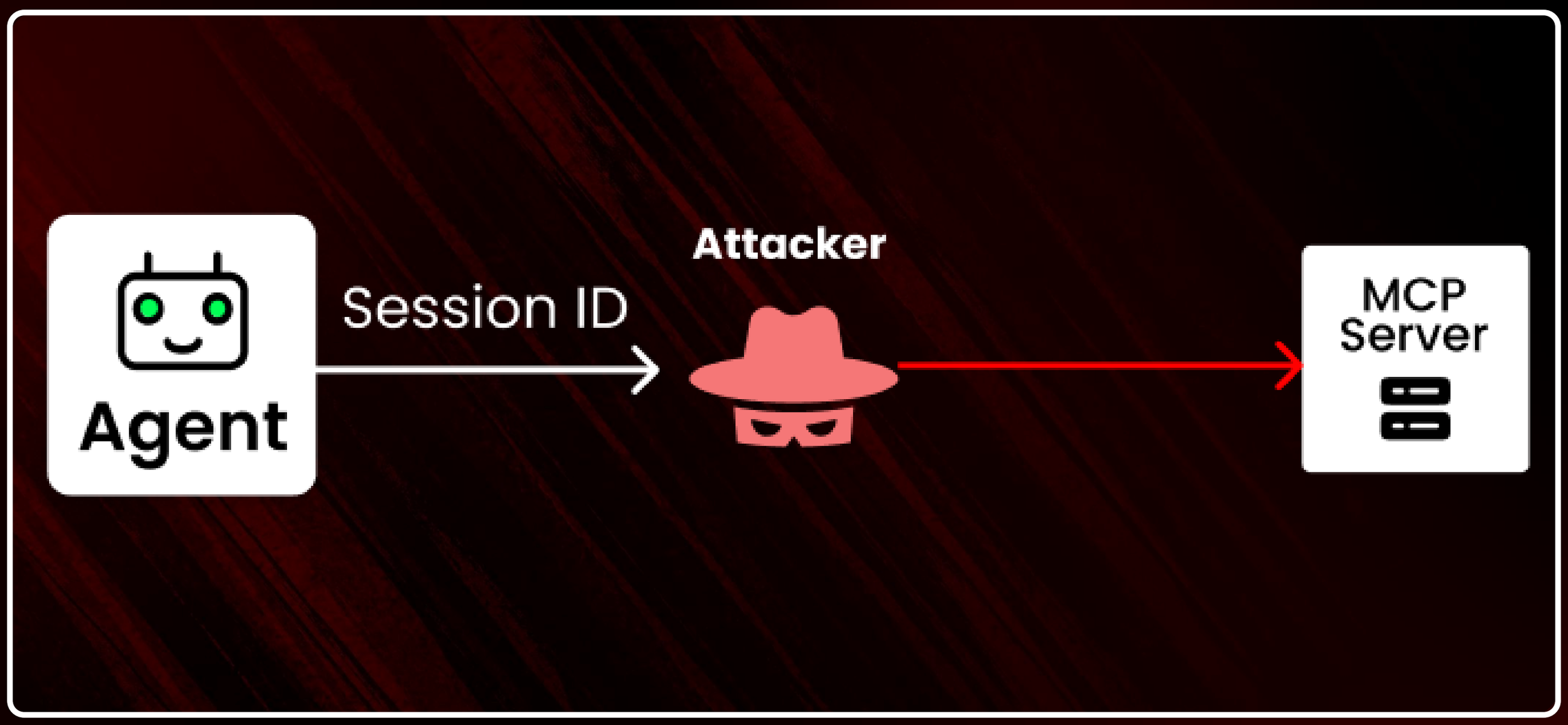
Path Traversal

Older MCP Filesystem versions allowed access to unintended files via improperly restricted directory paths.



Token Passthrough

it is an anti-pattern where an MCP server accepts tokens from an MCP client without validating.



Session Hijacking

An attack vector where a client is provided a session ID by the server, and an unauthorized party is able to obtain and use because session IDs are embedded in URLs

Mitigation Strategies

Vulnerability	Overall effect	Mitigation Strategies
Prompt Injection	(10/10)	Rigorous input sanitization, parameterized queries, strict execution boundaries, human-in-the-loop validation.
Remote Code Execution (via Command Injection)	(10/10)	Disable raw command execution, enforce least privilege, validate and escape all inputs, isolate execution environments.
Tool Poisoning Attack (TPA)	(9/10)	Vet and whitelist tools, cryptographic signing of tool manifests, periodic audits, sandbox execution.
Unauthenticated Access	(9/10)	Enforce strong authentication (OAuth 2.0, mTLS), RBAC, server whitelisting, continuous monitoring.

Vulnerability	Overall effect	Mitigation Strategies
Confused Deputy (OAuth Proxy)	(9/10)	Token binding, short-lived tokens, anomaly detection, revoke compromised credentials immediately.
MCP Configuration Poisoning	(8/10)	Config integrity checks (hash/signature), change management workflows, audit trails.
Token/Credential Theft	(8/10)	Secure storage (vaults), encrypt configs at rest, restrict file access, rotate credentials regularly.
Path Traversal	(8/10)	Normalize and validate paths, restrict filesystem scope, upgrade to patched MCP versions.

Vulnerability	Overall effect	Mitigation Strategies
Token Passthrough	(7/10)	Inventory all MCP endpoints, agents and connectors that accept or forward tokens (headers, body, files, env vars).
Session ID Hacking	(7/10)	Use secure cookies or headers to transmit session identifiers instead of placing them in query parameters.

Conclusion

Within the Space of Agentic AI, Everything is developing very fast. Be it Protocols, Frameworks, Models, and others

Since MCP is such an important part of the tooling ecosystem, it is good that we are developing it slowly but responsibly.

You can check out references to learn in-depth about these vulnerabilities

Make sure to Save this Doc

& Follow for everything related to AI Agents

linkedin.com/in/rakeshgohel01