ON2IT AND AMS-IX PRESENT: THREAT TALKS

Threat Talks

Zero Trust

in an agentic AI world

Zero Trust doesn't go away in an agentic AI world. It gets a new playing field. Instead of a handful of users and devices, you suddenly have swarms of AI agents acting, deciding, and connecting at machine speed. Same principles, way more moving parts.

Which raises the real question: how do you stay in control when software starts acting on its own?

Securing identities, behavior, and data in real time becomes critical. That's where things like hardware enforced privacy and verifiable AI execution come in, like what Confer is doing with its private AI protocol.

threat-talks.com

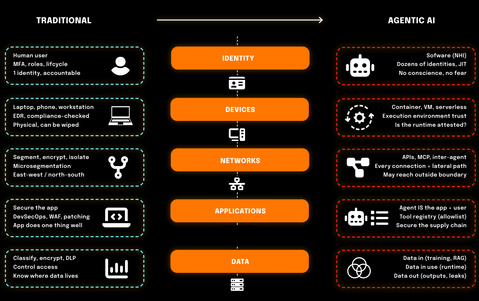
In this Threat Talks infographic we discuss:

- Zero Trust Pillars / Capabilities: Traditional versus Agentic AI
- The Agentic AI Attack Surface
- Confer

Zero Trust Pillars/Capabilities

Traditional vs. Agentic AI

Zero Trust was designed for a world of known users, managed devices, and relatively stable environments. You could map identities, enforce access, and monitor behavior with clear boundaries. In an agentic AI world, those boundaries blur. Identities become ephemeral, workloads spin up and down constantly, and actions happen without direct human input. The pillars don't change, but how and where you enforce them does.

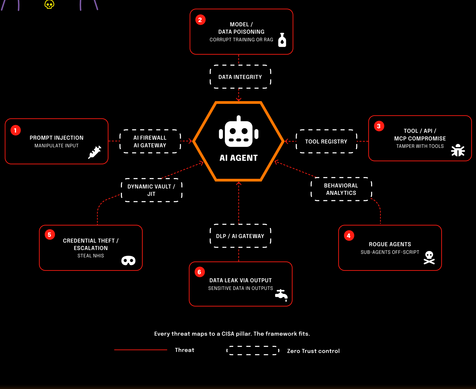


The strategy stays. The implementation changes.

The Agentic Attack Surface

Threats and Zero Trust Controls for AI Agents

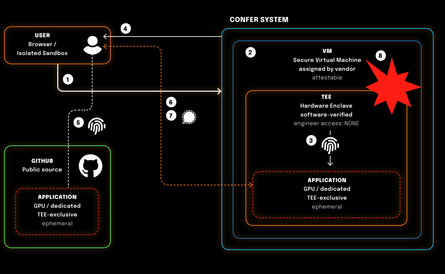
Agentic AI doesn't just increase risk, it reshapes it. Every agent can interact with APIs, datasets, tools, and even other agents, creating a web of dependencies that's hard to see and harder to control. That opens the door to new attack paths like prompt injection, data poisoning, and tool abuse, all happening at machine speed. The result is an attack surface that is wider, faster, and far less predictable.



Confer

End-to-end privacy, backed by cryptographic proof

Confer takes a different approach to trust. Instead of relying on policies or promises, it enforces privacy at the hardware level. Each session runs in a clean, isolated environment that is cryptographically verified before use, with end to end encryption that ensures only the user and the AI can access the data. No logs, no leftovers, no exposure. It is a model built for a world where trust needs to be proven, not assumed.



TLS Connection

1. The user opens confer.ai. A standard TLS connection is established – host verification, certificate pinning, encrypted channel. This is the same trust model as online banking. Nothing novel, nothing to trust beyond what your browser already validates.

VM + TEE Provisioning

2. Confer spins up a fresh, isolated Virtual Machine. Inside it, a Trusted Execution Environment (TEE) is initialised and loaded with a specific, versioned build of the Confer software. This is a clean-room environment – no persistent state, no prior session data, no other software.

Hardware Signs and Locks the TEE

3. The CPU's hardware security chip locks the TEE and generates cryptographic attestation: a signed proof of exactly what code is running, on what hardware, with what configuration. This signature cannot be forged – it comes from silicon, not software.

Attestation Delivered to User

4. The attestation package is sent back to the user's device: signed, cryptographic evidence of the running code and hardware. Your device receives proof before any AI conversation begins.

User Verifies (optional, but powerful)

5. The attestation can be independently verified against Confer's publicly available source code on GitHub. Anyone can download the source, compile it locally, and check that the resulting hash matches what is running on the server. You do not have to trust Confer – you can verify independently.

Noise Tunnel - Key Exchange

6. A Noise Protocol tunnel is established between the user's isolated browser sandbox and the TEE directly. Keys are exchanged end-to-end – the tunnel terminates inside the TEE, not at Confer's outer servers. No one between the user and the TEE can read the traffic.

Secure Chat Session

7. The session is now functionally equivalent to a Signal conversation. All messages are encrypted end-to-end through the Noise tunnel. The AI processes your prompt exclusively inside the TEE, using memory that is hardware-guaranteed to be inaccessible to anyone else – including Confer's own engineers. No logging. No side copies. No training data.

Session Ends - Everything Destroyed

8. When the session ends, the VM is torn down and all memory wiped. Ephemeral session keys are discarded – they exist nowhere. Even if a flawed TEE implementation left residual memory fragments, that memory is encrypted with keys that no longer exist. There is nothing to recover, nothing to subpoena, nothing to breach.

threat-talks.com

ON2IT ZERO TRUST INNOVATORS

amsix

Contact us @ team@threat-talks.com