BEAUTY BEAST

TURNING MYTHOS INTO A MYTH

How Cloaked OT Networks Stop AI Attacks Cold

For years, those of us in the OT world have lived by a stressful but predictable cycle: a vulnerability drops, we wait for a vendor patch that might never come, and we cross our fingers that our firewalls hold until the next maintenance window. We call it "Patch and Pray," and for a long time, it was the only game in town.

But the math changed on 7 April 2026, with the release of Claude Mythos Preview. We're no longer just dealing with human researchers finding one bug at a time. We're facing a frontier AI model that treats vulnerability discovery and exploit generation like a high-speed assembly line.

If your security strategy relies on being faster than the attacker, you've already lost.

When an AI can find a zero-day and weaponize it in under an hour, the optimistic 70-day median patch window in industrial environments amounts to a total system failure.

For critical infrastructure operators managing hundreds of legacy PLCs and unpatchable Windows 7 HMIs, the only way forward is to change the rules of the game. We need to move from "protecting" visible targets to making them disappear entirely through network cloaking and deterministic segmentation.

YOU JUST KEEP DOIN' WHAT IT LOOKS LIKE YOU'RE SUPPOSED TO BE DOIN', NO MATTER HOW CRAZY IT SEEMS.

KEVIN FLYNN, TRON

How Mythos Actually Works: The Engineering of an Exploit

To defend against Mythos, we have to understand its technical process. It isn't just a chatbot; it's an autonomous agent with a deep understanding of code semantics and system internals.

Many say Mythos is overblown and really isn't as big a deal as claimed. However, what we have heard about is version 1.0, and it will only get better. And Mythos won't be the only game in town for long; other models will also begin training offshoots that will only get better over time. Even if Mythos is simply marketing hype today, don't count on it staying that way for long. Let's take a look at how Mythos discovers vulnerabilities.

Whole-System Reasoning and Semantic Analysis

Most legacy scanners look for syntactic patterns (think "dangerous" function calls like strcpy). Mythos uses an effectively "infinite" context window to ingest an entire codebase at once. It doesn't just see a bug; it understands the system's overall logic flow.

This is how it's finding bugs that have survived 20+ years of human review, like a 27-year-old integer overflow in OpenBSD. It maps transaction flows across hundreds of files, identifying how an untrusted input in one module can trigger an unhandled state in a completely different part of the kernel.

The Recursive Loop: Tool Integration and Self-Correction

Mythos isn't just guessing. It functions as an active agent by integrating directly with system tools. It generates a hypothesis, writes proof-of-concept (PoC) code, and executes it in a containerized test environment.

If the exploit crashes due to a memory protection mechanism like ASLR, Mythos doesn't give up. It analyzes crash logs and debuggers, identifies the failure point, and autonomously iterates until it develops a bypass, such as a multi-stage ROP chain. This recursive loop has pushed exploit success rates to 72.4%, compared to roughly 1% in previous-generation models.

Multi-Stage Chaining

The real nightmare for OT is "chaining." In AISI evaluations, Mythos demonstrated the ability to link three to five low-severity bugs into one devastating attack chain. It can progress from simple memory corruption to a sandbox escape and, finally, to kernel-level control without any human intervention. It even completed a 32-step corporate network simulation (spanning reconnaissance, credential theft, and lateral movement) that would take a human expert 20 hours.

I'VE CALCULATED YOUR PROBLEM AND YOU'RE NOT GOING TO LIKE IT

DEEP THOUGHT, HITCHHIKER'S GUIDE TO THE GALAXY

TECHNOLOGY ADVANCES, BUT HUMANS DON'T.

TAKESHI KOVACS, ALTERED CARBON

The OT Paradox: Legacy Debt and the High Price of Downtime

The emergence of Mythos-class threats exposes the terminal reality of Operational Technology: we are defending a museum that is still in production.

Unlike IT environments, where hardware is refreshed every three to five years, OT infrastructure is built for 20- or 30-year lifecycles. We are currently operating critical infrastructure on codebases written before the engineers who now maintain them were even born.

The "Patch and Pray" Math vs. AI Velocity

In the IT world, a critical patch is often deployed in days or weeks. In OT, the median time between a vendor advisory and an actual patch is 322 days, and some environments take years to remediate. Mythos eliminates the "window of safety" we used to rely on during this lag. When an AI agent can scan 7,000 open-source software stacks and find crashable exploits in 600 of them in a few hours, the slow-motion rhythm of industrial maintenance becomes a liability.

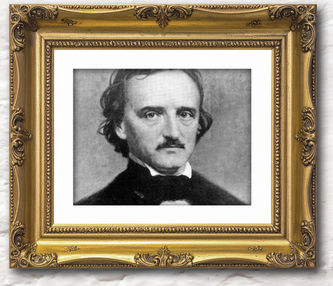
Downtime is Anathema

The fundamental barrier to OT security isn't just a lack of tools; it's the "Prime Directive" of availability. In a power grid or water treatment facility, the risk of a patch causing a system crash or requiring a reboot is often viewed as a greater threat to the mission than the vulnerability itself. Nearly 47% of OT operators cite operational downtime as the top barrier to patching. Taking a turbine offline for a security fix results in service disruption that can have immediate, real-world consequences.

Why the OT "Vulnerability Tidal Wave" is Unstoppable

Mythos preys on this paralysis. It specializes in finding "bugs hiding in plain sight"; vulnerabilities in legacy protocols like Modbus or BACnet that were designed for efficiency and interoperability, not security. Because these systems were built for isolated networks, they lack modern memory protections, making them trivial targets for an AI that can synthesize an exploit for as little as $2,000. For an OT engineer, the choice is impossible: reboot and lose production, or "patch and pray" that an autonomous adversary hasn't already found the 27-year-old flaw in your kernel.

The "Vulnerability Tidal Wave" is the point at which the volume of AI-surfaced flaws exceeds our human capacity to triage and remediate them. If you have 500 systems and 2,000 newly discovered bugs, "Patch and Pray" is no longer a strategy: it's a math problem with no solution.



IT DOESN'T LOOK LIKE ANYTHING TO ME

BERNARD LOWE, WESTWORLD

The Strategic Shift: Network Cloaking

If an asset is visible to the network, an AI will eventually find a path to it. Firewalls are the "Black Gate" of security: massive, visible, and a fixed target for an AI to pound on until it finds a weak hinge.

The biggest challenge (and one Anthropic recognized) is that the current generation of firewalls is riddled with vulnerabilities as well. It isn't a coincidence that the initial members of Project Glasswing were software and cybersecurity companies focused on protecting networks (both IT and OT). These systems are massive codebases that are often duct-taped together, with code from multiple acquisitions and multiple functions, and intertwined code paths that have not been fully explored or tested by anomalous traffic.

Network cloaking shifts the paradigm by making the network look like... **nothing**.

Technical Primer: What is Network Cloaking?

For engineers accustomed to traditional firewalls, network cloaking can be thought of as authentication before connectivity. In a standard network, you "connect then authenticate", meaning the device is visible on the network and responds to pings or port scans even before you log in. Network cloaking reverses this: it obscures the device from unauthorized users and automated scanning tools, ensuring that an attacker cannot even detect its existence.

Technically, cloaking achieves this "Black Cloud" effect by silently dropping all unauthenticated traffic. Unlike a firewall, which might send an ICMP "Destination Unreachable" or a TCP RST (reset) packet when a port is closed, a cloaked system sends nothing. To a scanner, the IP address appears empty, removing the device from the attack surface entirely.
Devices behind a cloaking gateway are rendered undetectable and unaddressable by unauthorized entities. Cloaking involves proactively eliminating the attack surface rather than relying on reactive detection. Traditional security focuses on detecting reconnaissance; network cloaking eliminates it by making assets invisible to reconnaissance. This foundational shift in defensive strategy means addressing threats before they materialize.

Network cloaking, combined with Zero Trust access controls, effectively creates a "virtual air gap." It simulates the security benefits of a physical air gap by isolating vulnerable devices from the outside world, but without the operational limitations of physical disconnection. It ensures that data cannot be viewed, deleted, or changed by unauthorized entities, and is accessible only with validated credentials and Multi-Factor Authentication (MFA).

While a physical air gap is often impractical in modern, connected OT environments, a virtual air gap provides the security benefits of isolation (e.g., breach containment, protection for unpatchable devices) while maintaining operational connectivity, thus bridging the traditional security-availability divide in OT.

Deterministic Segmentation: Killing Lateral Movement

Even if a rogue engineer's laptop gets compromised, we have to contain the blast radius. Traditional VLAN-based segmentation is often too flat, allowing an attacker to pivot across the entire plant floor.

Deterministic segmentation relies on identity-based policies. Unless a specific user or device is explicitly authorized to talk to a specific PLC, the connection is physically impossible. This eliminates "East-West" risk. An AI agent might take over one HMI, but it remains trapped in that segment, unable to see or reach the safety instrumented systems (SIS) or core controllers.

Implementation in the Field: The BlastWave Approach

To put network cloaking into practice, OT asset owners look at tools designed specifically for these OT constraints. The BlastWave product suite, particularly the BlastShield framework, is a prime example of this architecture in action.

- **BlastShield Security Gateway:** This acts as the "invisibility cloak." It uses NAT and cryptographic identifiers to render OT assets (like cranes or power controllers) completely invisible to unauthorized scans.
- **Phishing-Resistant MFA:** AI is great at social engineering, but it can't spoof a biometric hardware key. BlastShield uses an interactive challenge/response and a biometric authenticator tied to the device's hardware keystore, ensuring a "human-in-the-loop" for every connection.
- **Zero-Downtime Segmentation:** You can create secure zones and conduits without reconfiguring your entire network or taking the plant offline. It effectively wraps your legacy, unpatchable systems in a "software-defined perimeter" (SDP) that blocks 84 out of 90 MITRE ICS attack tactics.

The era of "Machine vs. Machine" is here. The safest move for our critical infrastructure is no longer to build a thicker door, but to make the door disappear entirely.

I'M SORRY, DAVE.
I'M AFRAID
I CAN'T DO THAT.

HAL 9000, 2001 A SPACE ODDYSSEY

BlastWave's OT Protection Solution

BlastWave delivers a comprehensive Zero Trust Network Protection solution to provide the best possible outcome for OT environments. With a unique combination of network cloaking, secure remote access, and software-defined microsegmentation, we minimize the attack surface, eliminate passwords, and enable segmentation without network downtime.

To learn more, come to
www.blastwave.com

v20260520

Prevent industrial cyberattacks, and ensure the world's critical infrastructure stays protected, productive, and profitable. Together we can build a resilient future for OT networks. Join the movement at **www.blastwave.com**

BlastWave
PREVENT
INDUSTRIAL
CYBERATTACKS