**Why the Trump Administration Views DeepSeek AI as a National Security Threat**

**Cyber Warfare Center Pacific (CYWAR-CENPAC)**

**Liberty Station, Point Loma CA 25 April 2025 12:42 PM EST**

In an era of great power competition, artificial intelligence is not just a tool—it's a weapon. The Trump administration has identified **DeepSeek AI**, China's rapidly ascending open-source AI powerhouse, as a **strategic threat to U.S. national security**. Here's why.

## 1. Military and Intelligence Leverage Through Dual-Use AI

At the heart of the concern lies DeepSeek's dual-use capabilities. While marketed as an open-source alternative for developers and businesses, **the underlying architecture of DeepSeek-V3 is ideally suited for military intelligence tasks**—including signals intelligence (SIGINT), autonomous targeting, cyber warfare, and real-time battlefield decision-making.

> "DeepSeek isn't just a chatbot—it's a tactical multiplier for the People's Liberation Army," one senior National Security Council official stated. "This is an LLM with a warfighting doctrine baked into its design."
>
> DeepSeek's focus on mathematics, coding, and long-context reasoning gives Chinese agencies a **cheap, scalable advantage in electronic warfare and battlefield command simulations**, eroding U.S. military superiority.

## 2. Geopolitical Soft Power via Open-Source Diplomacy

China's open-source play isn't altruistic—it's strategic.
By releasing high-performance models like DeepSeek-V3 into the global developer ecosystem, **Beijing is building loyalty among Global South nations, tech startups, and academic institutions** that feel locked out of American AI platforms due to licensing costs, censorship, or access restrictions.

> “This is AI Belt and Road,” says former CIA technical analyst Josh Winters. “They're creating digital dependencies and data pipelines under the guise of ‘open access.’”

DeepSeek is already being integrated into cars, robotics, surveillance systems, and digital infrastructure across Africa, Southeast Asia, and Latin America—**embedding Chinese-language AI as the default interface** in emerging markets.

### 3. Data Sovereignty Violations and Espionage Vectors

Recent revelations by South Korean regulators that DeepSeek **exfiltrated user prompts and metadata without consent** have ignited fears of backdoor data collection. U.S. intelligence believes this isn't accidental—it’s engineered.

> “This is not a bug. It’s a feature,” said Director of National Intelligence Tulsi Gabbard during a closed-door intelligence briefing.

The Trump administration suspects DeepSeek’s app architecture and API telemetry are **designed to harvest data across borders**, potentially feeding Chinese state repositories that inform behavioral surveillance, influence operations, and blackmail databases.
The National Security Agency (NSA) has traced anomalies in DeepSeek-related traffic that **mirror patterns observed in Chinese APT (Advanced Persistent Threat) groups**, further fueling espionage concerns.

### 4. Bypassing U.S. Export Controls via AI Inference Efficiency

China’s pivot to inference-optimized models is no coincidence.
By designing models that run effectively on older, locally available GPUs, DeepSeek enables **Chinese state and corporate actors to sidestep U.S. sanctions and hardware restrictions** imposed under the Export Control Reform Act and CHIPS Act.

> “They’ve built an AI system that thrives in an embargo,” said Commerce Secretary Peter Navarro. “It’s a shadow economy in code.”

This inversion—focusing on inference rather than training—undermines U.S. efforts to constrain China’s access to cutting-edge AI infrastructure.

The DeepSeek source code is located here: https://github.com/deepseek-ai/DeepSeek-V3/tree/main

Based on a comprehensive analysis of the **DeepSeek-V3.pdf** source code documentation, there is **no explicit evidence** of data collection, telemetry, logging, or hidden information pipelines intended for unauthorized user surveillance or data transmission to external servers (e.g., the PRC or elsewhere). However, there are **a few architectural components** and communication mechanisms that **could be adapted** or exploited for such purposes if DeepSeek's platform were modified for data exfiltration. Here are the relevant areas worth scrutiny:

### 1. Cross-Node Communication Pipelines

**Description:** DeepSeek-V3 employs high-efficiency **all-to-all cross-node communication** via InfiniBand (IB) and NVLink interconnects during both training and inference stages.

**Red Flag:** These are *ideal vectors for internal information sharing and coordination*, and if telemetry code were embedded at the kernel level or within the framework managing dispatch/combine kernels, it would be extremely difficult to detect without inspecting the compiled binaries and runtime traffic.

**Evidence:**

> "We customize efficient cross-node all-to-all communication kernels (including dispatching and combining)... For each token, when its routing decision is made, it will first be transmitted via IB... then forwarded via NVLink..."

### 2. Dynamic Routing and Redundant Expert Deployment

**Description:** During inference, DeepSeek dynamically routes tokens to multiple experts (up to 13 per token) using a combination of **static and on-the-fly routing algorithms**.

**Red Flag:** This form of expert-based routing could theoretically double as a **covert channel**, especially with redundancy and per-GPU load balancing designed to adjust in real time based on token patterns.

**Evidence:**

> “Each GPU hosts more experts (e.g., 16 experts), but only 9 will be activated during each inference step… we compute the globally optimal routing scheme on the fly.”

### 3. MTP Modules Discarded During Inference

**Description:** The **Multi-Token Prediction (MTP)** modules are used during training but discarded in deployment.
**Red Flag:** Transient modules that are active during training and not included in inference are harder to inspect by the public. These MTP modules could be exploited for data staging or internal telemetry.
**Evidence:**

> “During inference, we can directly discard the MTP modules… we can also repurpose these MTP modules for speculative decoding…”

### 4. Lack of Details on External API Interactions

There is **no mention in the PDF** of logging, telemetry endpoints, external analytics, or persistent connections to outside servers. That said, the **DeepSeek website and platform endpoints (chat.deepseek.com, platform.deepseek.com)** are proprietary and **not covered in the technical PDF**, so it's possible those endpoints **could include telemetry or data tracking features**.

**Summary**

- **No definitive proof** of spyware or surveillance in the open-source architecture.
- Several **suspicious or high-risk architectural components** (cross-node comms, dynamic routing, dropped modules) exist that *could* be repurposed for covert data transmission.
- Actual threat assessment would require **runtime inspection, packet sniffing, and binary analysis**—not just the open-source documentation.
- **Closed API endpoints and web platform behavior** remain a **black box**.

**Technical Security Audit Plan for DeepSeek AI (V3 Series)** to identify hidden back-end information pipelines, unauthorized data collection, and transmission to external servers (e.g., PRC-controlled infrastructure):

**DeepSeek-V3 Security Audit Plan**

**Objective:** Detect, analyze, and mitigate any covert telemetry, data exfiltration, or unauthorized data collection mechanisms in the DeepSeek-V3 model—whether embedded in source code, model weights, or API interactions.

## 1. Scope of Audit

**Systems to Examine:**

- Local deployments (open-source GitHub + Hugging Face weights)
- API-based deployment at `platform.deepseek.com`
- Web-based UI at `chat.deepseek.com`
- Associated back-end containers, packages, and kernel modules
- Any integrated SDKs, binaries, or hardware accelerators (NVIDIA/AMD/Huawei Ascend)

## 2. Audit Toolchain

LayerTools/Techniques

**Static Code** `grep`, Semgrep, Bandit, Ghidra, Binwalk

**Network Monitoring** Wireshark, Zeek, tcpdump, mitmproxy

**Runtime Behavior**s trace, lsof, valgrind, AppArmor/auditd

**Container Inspection** DockerSlim, Clair, Trivy

**Model Analysis** ONNX Parser, PyTorch Inspector, FP8/BF16 cast validators

**Binary Reverse Engineering** Ghidra, IDA Pro  (proprietary binaries used)

## 3. Audit Phases

### Phase 1: Source Code Inspection (Open-Source GitHub)

- Searched for hardcoded URLs, IP addresses, or domain names (e.g. `.cn`, `.gov.cn`, `.ali`, `.baidu`, `.cmcc`)
- Identified use of:
- `requests`, `http.client`, `socket`, `subprocess`
- `os.system`, `eval`, `exec`

- Search for hidden `.env`, telemetry flags, or unknown callbacks
- Examined:
- `generate.py` (core inference script)
- `convert.py` (model weight transformation logic)
- Any config templates for routing or GPU comms

**Red Flags:**

- Hidden logging to 3rd-party domains
- Dynamic routing config loaded from external sources

**Phase 2: Network Traffic Monitoring**

- Used Wireshark/Zeek to:
- Record traffic during inference runs (including first-time setup)
- Look for DNS lookups or traffic to China-based ASNs
- Capture headers for all HTTPS requests
- Run on isolated testbed with known IP whitelisting

**Red Flags:**

- Unexpected outbound traffic during inference
- All interactions with `*.cn`, `*.deepseek.ai`, or `cmcc.com`

**Phase 3: Model Weights and Binary Inspection**

- Verify hash of downloaded weights
- Decompile:
- FP8 model conversion scripts
- Embedded binaries from `sglang`, `triton`, `lmdeploy`, etc.
- Examined MTP modules and discarded training components

**Red Flags:**

- Hidden payloads embedded in weight files (used Binwalk)
- Model behaviors that vary when tokens suggest PII, military, or legal content

**Phase 4: API Endpoint Behavior**

- Used `mitmproxy` and `Burp Suite` to intercept API calls to `platform.deepseek.com`
- Test:
- User prompt → model completion → inspected headers and body
- Analyzed any cookies, session tokens, or device fingerprints

**Red Flags:**

- Transmission of prompt text in plaintext
- Session tracking identifiers without user consent

**Phase 5: Hardware and Driver-Level Telemetry**

- If running on Huawei or Chinese-manufactured accelerators (e.g., Ascend NPU):
- Analyzed device drivers for background daemons
- Used `auditd` and `strace` to log system calls
- Blocked outbound connections with `iptables` and log attempted violations

**Red Flags:**

- Calls to hardware telemetry endpoints
- Drivers attempting cloud-based updates

**4. Reporting**

- Maintained a detailed **finding log** with timestamps, tools used, and suspected behavior
- Flag **confirmed violations** of:
- GDPR, U.S. Federal Privacy Regulations
- Export Control Laws (e.g., EAR)
- DOD/IC data sovereignty policies

**5. Containment & Response Strategy**

If a malicious data exfiltration pipeline is confirmed:

- Revoke any instances using DeepSeek in national security, defense, or legal contexts
- Blacklist IPs and domains used in outbound comms

- Notify the DOJ and ODNI (if classified info is involved)
- Issue an embargo directive to contractors via DHS and CISA.