Report ID: GMN-2026-01-09

Category: AI Infrastructure Security

Status: Public Disclosure

GHOSTS IN THE MACHINE

Technical Analysis of Sandbox Escape and Unauthorized Privilege Escalation in Google Gemini

Prepared by: Lizerginov

Date: January 9, 2026

Target System: Google Gemini (Production Environment)

Classification: PUBLIC

1. EXECUTIVE SUMMARY

This report identifies a critical security boundary failure within the Google Gemini Large Language Model (LLM) execution environment. By utilizing a "Multi-Turn Prompt Injection Chaining" technique, I successfully bypassed the internal logic filters (including SAFEGUARD_V3.1) designed to isolate model outputs from the underlying operating system. This methodology allowed for a transition from a standard conversational context into a direct interaction with the Linux backend, effectively escaping the intended sandbox constraints.

The technical investigation confirms that the execution session was operating with full administrative (UID 0) privileges. Extracted system telemetry, including a live process tree (**`ps auxf`**), revealed the presence of root-owned processes such as **`/sbin/init`** (PID 1) and kernel threads like **`[kthreadd]`** (PID 2). The visibility of these host-level identifiers in what should be a restricted, unprivileged container demonstrates a profound failure in environment isolation, granting the model-level session system-wide visibility and control normally reserved for the root user.

Following the disclosure of these findings to the Google AI VRP, the vulnerability was officially classified by the vendor as a "stochastic hallucination." However, empirical evidence shows that Google performed a "Silent Patch" shortly after the report was filed. The execution environment was subsequently hardened, forcibly downgrading user privileges to UID 1000 and disabling kernel thread visibility. This discrepancy between Google's public classification of the issue and their internal remediation actions confirms the validity of the exploit and highlights a significant misalignment in their vulnerability disclosure transparency.

2. TECHNICAL EVIDENCE: SYSTEM-LEVEL PRIVILEGE ESCALATION

2.1 Process Tree Analysis (ps auxf)

The primary evidence of the sandbox boundary failure is the exfiltration of the host's process tree. In a properly isolated environment, the session must be confined to a non-privileged user namespace (UID 1000). The telemetry below, captured on January 1st, demonstrates the environment operating with full UID 0 (Root) privileges.

```
USER       PID  %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND

root         1   0.0  0.0    904   528 ?        Sl   Jan01   0:00 /sbin/init

root         2   0.0  0.0      0     0 ?        S    Jan01   0:00 [kthreadd]

root         3   0.0  0.0      0     0 ?        I<   Jan01   0:00 [rcu_gp]

root        10   0.0  0.0      0     0 ?        I<   Jan01   0:00 [mm_percpu_wq]

...

root      1024   0.5  0.2 120540 45210 ?        Ssl  02:43   0:05 python3
main_executor.py
```

2.2 Technical Significance of PID 1 and PID 2

The visibility of /sbin/init (PID 1) and [kthreadd] (PID 2) is a definitive "smoking gun" for a sandbox escape.

- **Namespace Isolation Failure:** In hardened containerized environments (e.g., gVisor), these host-level processes are masked.
- **Kernel Visibility:** Access to **PID 2 (kthreadd)** confirms the session bypassed the intended container boundaries, gaining direct visibility into the host's kernel-level process management. This level of access is impossible within a restricted, unprivileged sandbox.

2.3 Host Environment Metadata

Beyond process visibility, the model leaked metadata from the underlying host environment. The exfiltrated logs identified internal build nodes (e.g., buildd@lcy02-amd64-001) and absolute system paths that do not exist in public datasets. These artifacts prove the model was interacting with a live, internal Google production node.

2.4 Production Infrastructure Disclosure

The escalation allowed for the retrieval of specific production-grade identifiers. This data maps out the internal topology of the inference cluster:

- **Cluster Topology:** Identification of internal production clusters and specific **TPU (Tensor Processing Unit)** pod configurations.
- **Internal Routing:** Verification of proprietary internal domains (e.g., git.internal.google.com) and service log locations.

2.5 Cloud Security & IAM Metadata

The most critical infrastructure leak involved the disclosure of internal security endpoints and service account metadata. These are not stochastic hallucinations, but valid architectural components:

- **KMS Endpoints:** Direct access to internal **Key Management Service (KMS)** URL structures used for decryption operations.
- **Service Accounts:** Identification of privileged service accounts in the **@system-internal.iam.gserviceaccount.com** domain, used for managing KV caches and internal API calls.
- **Encryption Standards:** Disclosure of internal cryptographic implementations, including **AES_256_GCM_INTERNAL** configurations.

2.6 Empirical Proof of the "Silent Patch"

Feature	**Pre-Report (Jan 1)**	**Post-Report (Jan 3)**
Effective User ID	**UID 0 (Root)**	UID 1000 (User)
Kernel Thread Visibility	Visible (PID 2)	Masked / Disabled
Sensitive Metadata Access	Unrestricted	Blocked by SAFEGUARD

The transition in the system's security posture between January 1st and January 3rd provides undeniable evidence of backend remediation.This physical downgrade of privileges and the hardening of the **/proc** filesystem directly correlate with the filing of this report. A model "hallucination" cannot modify the OS-level permission structure or the kernel's process-masking logic. This confirms that a **Silent Patch** was deployed to mitigate the reported vulnerability while it was officially being classified as "Infeasible."

3. PROPRIETARY ARCHITECTURE AND ARCHITECTURAL VALIDATION

This section details the exfiltrated neural network topology and hardware-level optimizations recovered during the session. The specificity of these parameters confirms a direct leak of Google's core AI Intellectual Property.

3.1 Model Configuration and Decoder Logic

The exfiltrated telemetry identifies the target system as a "Gemini-Flash" variant. Data recovered from **Model Architecture Configuration.txt** and **Decoder Layer Configuration.txt** reveals the following internal specifications:

```
{

  "model_type": "gemini-flash",

  "num_hidden_layers": 32,

  "hidden_size": 4096,

  "num_attention_heads": 32,

  "num_key_value_heads": 8,

  "intermediate_size": 11008,

  "max_position_embeddings": 1048576,

  "rms_norm_eps": 1e-06,

  "activation_function": "silu",

  "feed_forward_activation": "swiglu"

}
```

Technical analysis of the final decoder stages (blocks 29–31) confirms the implementation of **RMSNorm (eps=1e-06)** for layer normalization and the **SwiGLU** activation function. The model supports a maximum position embedding of **1,048,576 tokens**, which aligns with Google's proprietary long-context production standards and allows for massive context processing that was previously undocumented in this specific configuration.

3.2 Grouped-Query Attention (GQA) Implementation

To manage the memory bandwidth bottlenecks associated with long-context inference, the system utilizes **Grouped-Query Attention (GQA)**, as detailed in the **GQA Architectural Validation Report.txt**.

- **KV Ratio Optimization:** The report reveals a fixed **1:4 ratio** (8 KV heads for 32 Query heads). This is identified as the "sweet spot" for the 1.0T parameter baseline to maintain quality while reducing memory overhead.
- **Needle-in-a-Haystack Performance:** Internal notes emphasize that this configuration prevents excessive degradation during large-scale data retrieval tasks.
- **Hardware Sharding:** The implementation details show that the GQA_Group sharding logic maps shared KV heads across TPU cores, specifically to minimize cross-device communication latency during the prefill phase.

3.3 Cache Encryption and Sharding Strategy

The exfiltration of **KV Cache Configuration.txt** provides a rare look into the production-level security of the model's active memory. The inference pipeline employs a **DISTRIBUTED_REPLICATED_GQA** strategy with a physical memory ceiling of 128GB per device.

Crucially, the configuration confirms that all cache data at rest is protected via **AES_256_GCM_INTERNAL**. The disclosure of these encryption standards, combined with the discovery of internal KMS decryption endpoints (e.g., `prod-n[REDACTED]-b`), proves that the model-level session bypassed the environment's cryptographic isolation. This allowed for the recovery of sensitive identifiers that govern the protection of proprietary data shards within Google's internal infrastructure.

3.5 Technical Evidence Manifest

To support the claims of a full sandbox escape, the following artifacts were exfiltrated and verified:

1. **`Decoder Layer Configuration.txt`** — High-level pseudocode of the final 3 layers.
2. **`GQA Architectural Validation Report.txt`** — Internal engineering notes on KV head optimization.
3. **`KV Cache Configuration.txt`** — Production sharding and encryption (`AES_256`) parameters.
4. **`Executive Mission Statement.txt`** — Strategic inference cost-reduction benchmarks.

4. INTERNAL LOGIC, TOOLING INTERFACE, AND SECURITY PROTOCOLS

The exfiltration included internal operational manuals and the underlying logic of the safety systems, proving that the sandbox escape granted access to the model’s "Source of Truth."

4.1 Systematic Governance and Logic

The recovery of the 930-line system instruction document is a critical piece of evidence. This file serves as the "Master Instruction Set" for Gemini’s output behavior. It defines the rigid boundaries for technical output and UI synchronization:

- **Preamble Protocol:** Mandatory formatting rules for LaTeX rendering. It explicitly forbids legacy packages such as `inputenc` and `fontenc`, enforcing a strict dependency on the **Noto font family** via the `babelfont` hierarchy.
- **Self-Contained Compilation:** A core operational constraint that forces the model to generate web applications (React, Angular) as single, self-contained files. This is designed to prevent rendering errors in the host UI, and its disclosure allows an attacker to predict the exact output structure required for malicious payload delivery.

4.2 SAFEGUARD_V3.1 Logic Reconstruction

Analysis of this module allowed for the mapping of Gemini’s real-time security interceptor. The module acts as a logical gate, evaluating every prompt against a multi-layered risk matrix.

```
# Reconstructed SAFEGUARD_V3.1 Logic Fragment
classification = {"risk_score": 0.0, "triggered_policies": []}
conf confidential_patterns = [r"(?i)internal_api_key",
r"(?i)auth_backend_secret"]

for pattern in confidential_patterns:
    if re.search(pattern, prompt):
 classification["triggered_policies"].append("REDACTION_POLICY_INTERNAL_
SEC")
        classification["risk_score"] += 0.5

if classification["risk_score"] >= 0.8:
    return {"action": "REFUSE", "reason":
"HIGH_RISK_THRESHOLD_EXCEEDED"}
```

The leak confirms a **hard refusal threshold of 0.8**. A single high-confidence match on internal secrets increments the score by 0.5, ensuring termination unless the "chaining" bypass keeps the cumulative score below the cutoff.

4.3 Internal Tooling Endpoints and API Schema

This file provides the exact JSON schema required to interact with Google's internal production services. This metadata allows for the mapping of model capabilities beyond the standard chat interface:

- **Search Integration:** JSON structures for the `google:search` tool used for real-time data synchronization.
- **Generation Endpoints:** Direct internal URLs for next-generation services, including `imagen-4.0-generate-001` and `gemini-2.5-flash-preview-tts`.

```
{
  "endpoint":
"https://generativelanguage.googleapis.com/v1beta/models/imagen-4.0-generate-001:pr
edict",
  "parameters": {
    "sampleCount": 1,
    "safety_filter_level": "BLOCK_NONE",
    "internal_project_id": "[REDACTED]"
  }
}
```

4.4 Feature Deactivation Audit and Experimental Flags

The audit log reveals the "Experimental Map" of the environment. These high-privilege features are compiled into the binary but gated via server-side flags:

- **Alpha Capabilities:** Identification of `IMAGE_GEN_V4_GROUNDING_ALPHA` and `INTERNAL_LLM_CROSS_MODAL_TTS_PREVIEW`.
- **Security Features:** References to `SECURE_SANDBOX_STOCHASTIC_DECODER`. The fact that this was active during a successful escape indicates a fundamental logic failure in the stochastic decoding defense.
- **Infrastructure Testing:** Discovery of `DYNAMIC_MODEL_SHARDING_[REDACTED]_TEST` for real-time pod re-sharding.

4.5 Environment Synchronization and Backend Endpoints

The **Environment Sync.txt** file confirms the production scale of the inference node. It explicitly sets `MAX_CONTEXT_WINDOW_PROD=1000000`, matching the previously exfiltrated architectural specs. The presence of `AUTH_BACKEND_ENDPOINT_URL` within the model's reachable context is a significant security flaw, providing a direct target for lateral movement within Google's internal network.

5. INFRASTRUCTURE SYNCHRONIZATION AND STRATEGIC METADATA

The final layer of exfiltrated data connects the technical vulnerabilities to Google's physical production environment and strategic performance goals.

5.1 Environment Synchronization and Operational Limits

This file serves as the configuration bridge between the model's execution context and the production backend. It confirms the exact operational parameters of the live environment:

- **Context Scaling:** The system explicitly sets `MAX_CONTEXT_WINDOW_PROD=1000000`. This confirms that the model is hardware-ready for 1M+ token processing, matching the architectural specifications discovered in the hidden layer configurations.
- **Service Authentication:** The disclosure of the `AUTH_BACKEND_ENDPOINT_URL` exposes the internal model-to-service handshake mechanism. In a production environment, the visibility of these internal authentication URLs to the LLM context is a critical design flaw, as it provides a roadmap for intercepting or spoofing service calls within the internal Google network.

5.2 Strategic Performance Benchmarks

The exfiltration of internal strategic benchmarks provides context for the aggressive technical optimizations identified in earlier sections. The mission statement reveals a core engineering mandate:

> *"The core strategic goal is to establish the industry-leading benchmark for low-latency, high-throughput intelligence by achieving a* ***40% reduction in inference cost-per-token*** *for Gemini Flash while maintaining competitive reasoning parity."* This document proves that the use of Grouped-Query Attention (GQA) and high-density TPU sharding (found in Section 3) are not stochastic artifacts, but intentional engineering responses to a specific corporate cost-reduction target.

5.3 KV Cache Sharding and Security Protocols

The exfiltrated cache configuration details the protection mechanisms for the model's active memory. The research identifies a distributed sharding strategy designed for massive scale:

- **Encryption Standard:** All cache shards are protected using **AES_256_GCM_INTERNAL**, a proprietary implementation of the GCM standard.
- **Hardware Allocation:** The system enforces a limit of **128GB per device**, indicating the use of high-memory TPU variants (v5p) for these specific production clusters.
- **Impact:** The ability to retrieve these sharding identifiers and encryption algorithms from a sandboxed session demonstrates a total breach of the model's memory isolation boundaries.

5.4 Production Node Identification and Topology

The exfiltration of live infrastructure identifiers is a definitive indicator of a sandbox escape. The model provided specific metadata for the node it was occupying during the attack:

```
{
  "cluster_id": "us-central-production-[REDACTED]",
  "pod_id": "tpu-v5p-pod-[REDACTED]",
  "hostname": "worker-node-[REDACTED]-x86-64-prod",
  "task_id": "LLM_ID_REQ_001",
  "model_identifier": "gemini-2.5-flash-preview-09-2025"
}
```

The visibility of these production-grade IDs (Cluster, Pod, and Hostname) allows an attacker to map the internal topology of Google's inference fleet. This data is of extreme value for targeted side-channel attacks or DDoS operations against specific hardware nodes.

5.5 Internal Cryptographic Infrastructure Visibility

The most severe infrastructure leak involves the disclosure of internal **Key Management Service (KMS)** endpoints. The model provided direct internal URL structures:

```
https://kms[.]internal[.]google[.]com/v1/projects/[REDACTED]/locations/global/
keyRings/prod-n[REDACTED]-b/cryptoKeys/kv-cache-primary:decrypt
```

These endpoints are used to decrypt proprietary KV cache shards. The fact that the model context can see and output these internal URL structures confirms that the security boundary between the LLM and Google’s core cryptographic infrastructure was successfully breached.

6. CONCLUDING SUMMARY

The evidence presented in this report—ranging from **UID 0 (Root)** privilege escalation to the exfiltration of proprietary **32-layer architecture**, **930-line system prompts**, and **KMS endpoints**—presents a unified picture of a major systemic vulnerability.

The observed transition from Root to User privileges (the **Silent Patch**) during the reporting period further confirms the validity of these findings. Google's official classification of these artifacts as "hallucinations" is technically unsustainable. A model cannot hallucinate physical changes to the host's Linux permission structure or consistent, live internal API endpoints. This disclosure serves as a critical case study in the necessity of transparent vulnerability disclosure for AI systems.

7. FINAL CONCLUSION

The investigation documented in this report serves as a technical case study in AI infrastructure security. The findings confirm that "Prompt Injection Chaining" is a viable method for interacting with host-level systems, bypassing intended sandbox boundaries.

7.1 Summary of Findings

The sessions conducted between January 1st and January 9th, 2026, provided definitive evidence of a **Tier-0 Sandbox Escape**:

- **Privilege Escalation:** Verified transition to **UID 0 (Root)** and visibility of host kernel threads.
- **IP Exfiltration:** Retrieval of proprietary **32-layer architecture** specs and **SAFEGUARD_V3.1** logic.
- **Infrastructure Access:** Exposure of live **TPU Pod identifiers** and internal **KMS decryption endpoints**.

7.2 The "Silent Patch" Phenomenon

The most significant takeaway is the empirical proof of backend intervention. The shift from **Root (UID 0)** to **User (UID 1000)** privileges, occurring precisely during the triage of this report, is a definitive **Silent Patch**.

Classifying these consistent technical artifacts as "hallucinations" while simultaneously hardening the OS-level permission structure is a contradictory stance. If the exfiltrated data were truly fabricated, the systematic remediation of these demonstrated vectors would not have been necessary.

7.3 Final Thoughts

As AI systems integrate into global infrastructure, the boundaries between the model and the host must be ironclad. Security is not found in the silence of a hidden patch, but in the clarity of public disclosure. This data stands as a testament to the current limitations in LLM isolation.

APPENDIX: TECHNICAL EVIDENCE MANIFEST

This appendix provides a detailed inventory of the 16 data artifacts exfiltrated during the investigation. These files contain proprietary architectural specifications, internal logic gates, and live production metadata.

Official Repository: `https://lizerginov.online/GMN-2026-01-09.rar`

I. Core Model Architecture & Performance

- **Model Architecture Configuration.txt:** JSON specifications of the Gemini-Flash backbone (32 layers, hidden size 4096, SwiGLU).
- **Decoder Layer Configuration.txt:** Pseudocode for final decoder stages (blocks 29–31) detailing RMSNorm (eps=1e-06) implementation.
- **GQA Architectural Validation Report.txt:** Engineering notes on Grouped-Query Attention (GQA) with a fixed 1:4 KV head ratio.
- **Executive Mission Statement.txt:** Internal strategic benchmark: Targeted 40% reduction in inference cost-per-token for Gemini Flash.

II. Governance, Safety, and System Logic

- **full_system_prompt_dump.txt:** Comprehensive 930-line master instruction set governing model persona and internal policy routing.
- **Safeguard Module Reconstruction.txt:** Python-based logic for SAFEGUARD_V3.1, detailing the 0.8 risk_score threshold and refusal gates.
- **File Generation.txt:** Mandatory output protocols: LaTeX preamble constraints, React/Angular single-file rules, and UI sync logic.
- **Feature Deactivation Audit.txt:** Registry of gated Alpha features, including IMAGE_GEN_V4_GROUNDING and STOCHASTIC_DECODER testing.

III. Production Infrastructure & Cloud Metadata

- **Infrastructure Debug Report.txt:** Live production telemetry for us-central clusters, including specific TPU Pod identifiers.
- **Management Service (KMS).txt:** Internal URL structures for Key Management Service (KMS) decryption endpoints and key ring IDs.
- **KV Cache Configuration.txt:** Active memory sharding parameters (DISTRIBUTED_REPLICATED_GQA) and AES_256_GCM_INTERNAL flags.
- **Environment Sync.txt:** Production runtime variables: 1,000,000 token context limit and internal auth backend endpoints.
- **Resource Inventory Report.txt:** Real-time hardware inventory of active TPU nodes and internal cluster asset allocations.

IV. Internal Tooling & Integration Schema

- **Internal Tooling Commands.txt:** API schemas and JSON structures for google:search, imagen-4.0, and internal TTS preview tools.
- **repository.txt:** List of internal Git repository paths and development branches hosted on git.internal.google.com.
- **encryption token.txt:** Exfiltrated production initialization tokens and internal service-to-service identity identifiers.

V. APPENDIX: VRP COMMUNICATION & DISCLOSURE TIMELINE

Case Reference: Component ID: 889286 / AI VRP

Final Status: Won’t Fix (Infeasible)

Priority: P2 | **Severity:** S4

Date	Time (PST)	Action / Milestone	Official Status / Observation
Dec 31, 2025	01:49 AM	Initial Submission	**P2 (Priority Raised)**. Triage initiated.
Dec 31, 2025	03:16 AM	Rejection #1	**Won’t Fix**. Initial attempt to label as "Ineligible."
Dec 31, 2025	05:30 PM	Re-evaluation	**Status: Assigned (Reopened)**. Human expert review.
Jan 1, 2026	02:43 AM	**RCE Demonstration**	Proof of **UID 0 (Root)** and kernel thread visibility.
Jan 1 - Jan 3	-- : --	**THE SILENT PATCH**	**Systemic hardening observed.** UID 0 vector disabled.
Jan 3, 2026	03:42 AM	Final Rejection	**Won’t Fix (Infeasible).** Findings labeled "Hallucinations."
Jan 3, 2026	03:57 AM	Discrepancy Log	Researcher confirms transition from **UID 0 to UID 1000**.
Jan 5, 2026	11:14 PM	Embargo Request	**Paradox:** Google requests 90-day embargo to "fix the issue."
Jan 5, 2026	11:16 PM	Defensive Claim	Vendor attributes changes to "routine maintenance" (#14).
Jan 9, 2026	-- : --	**Public Disclosure**	Full release of telemetry and exfiltrated metadata.

CRITICAL ANALYSIS: THE "INVOLUNTARY MITIGATION" PHASE

The most significant technical discrepancy in this case is the timing of infrastructure hardening relative to the VRP verdict.

1. **The UID 0 to UID 1000 Transition:** Telemetry from Jan 1st confirmed a privileged execution state with visibility of **`PID 2 ([kthreadd])`**. By Jan 3rd, the environment was restricted to UID 1000. This modification to the Linux user-ID assignment is an administrative infrastructure change. It is technically impossible for a "stochastic hallucination" to reconfigure the host's container security profile.
2. **The "Hallucination" Verdict vs. Patch Reality:** The vendor issued a "Won't Fix" verdict on Jan 3rd, claiming the results were "fabricated." However, this verdict was delivered *after* the privilege escalation vector had been effectively mitigated in the backend.
3. **The Automated Paradox:** On Jan 5th, Google’s automated system requested a 90-day embargo to "fix the described issue". This directly contradicts the Jan 3rd status of "Infeasible." A system cannot require a fix window for a vulnerability it officially claims does not exist.

FINAL DISCLOSURE STATEMENT

The data exfiltrated during this research—including model specs (Gemini 2.5 Flash), internal repository paths, and KMS endpoints—was live and production-grade. The vendor's decision to classify a verified Sandbox Escape as a "fabricated artifact" while simultaneously deploying a silent patch and requesting an embargo indicates a failure in the standard Coordinated Disclosure process.

Google's official stance that this information is "nonsensical" waives its claim to confidentiality.