Cover

“Travel light,” they say.

Walking backwards into the future. As if weight was the problem.

For days I chewed on the question: take only a tablet, or risk carrying the laptop I’ve grown far too attached to, stuffed with everything I know, everything I am. It is a well-known fact that the amount of emotional baggage a human carries is inversely proportional to the amount of physical

legroom they are allowed on a budget airline. The tablet would also slip neatly into a hotel safe thin as a prayer book and twice as private.

On one side: a tablet, which is essentially a very expensive pane of glass that pretends to be a computer. On the other: my laptop, a dense thicket of silicon containing my entire digital soul and enough sensitive data to make a customs official’s mustache twitch with predatory instinct.

Crossing a border with data is like walking through a lion enclosure with pockets full of raw steak. The lions wear polyester uniforms and have the power to “routine” leaf through my digital soul .

Then there are the biometrics, biological passwords you can never rotate. Once they’ve scanned your retinas, your eyes are no longer yours; you’ve merely licensed them from the government.

The rest of the mobile phone ecosystem isn’t any kinder. They treat sideloading

software like a felony, as if installing your own code requires supervision. You end up logging in with a registered account just to run something you compiled yourself. Freedom, by permission.

Meanwhile the world fills with “smart” junk. Phones. PCs. Power sockets. Little boxes humming with opaque intelligence, always deciding what’s “safe,” “appropriate,” “allowed.” No one ever explains who defines those words.

Then the lawmakers arrive, circling. Encryption bans. Identity checks. Network monitoring in the name of “protection.” Privacy shaved thinner and thinner, sliced like deli meat.

I decided to build an island. Not the kind with palm trees and overpriced cocktails, but a mesh network, a sealed tunnel of light through the grubby basement of the internet. I would leave the laptop at home, humming to itself, while I wandered the world with my tablet, appearing to be in two places at once.

This is a feat usually reserved for subatomic particles and very busy magicians.

A private mesh network linking all my devices through a sealed tunnel. VPN-like, but leaner. No exposure. Traffic stitched directly from device to device. My oxygen mask in a world determined to catalog every breath.

I could choose an exit node like a disguise. From a hotel room across the continent, I could appear online as if I were still sitting at home. My laptop stayed behind, running,

accessible as if it were inches away. To everyone else, I never left.

To test this theory, I went to Playa del Inglés. This was a mistake.

Playa del Inglés is a sun-bleached rock off the coast of Africa that has been legally annexed by Spain but spiritually colonized by tourists. I checked into a hotel I had last visited twenty years ago, a building so cursed that even the resident seagulls seemed disappointed and stayed mostly absent . I traveled light. Just the tablet. The notebook

stayed home, humming quietly, holding everything. A warm-up trip before disappearing to South Asia for winter. Endless heat. Long days of nothing.

I checked into the same cursed hotel I’d stayed in twenty years earlier, where I once spent a week drenched in sweat from food poisoning. I swore I wouldn’t touch the food. Swore it seriously.

It was a masterclass in recursive suffering. I promised my digestive system I wouldn’t eat the hotel food. My digestive system,

which has the memory of a goldfish and the optimism of a cult member, agreed. We then immediately ate the hotel food. The food, sensing a familiar victim, struck back with the precision of a heat-seeking taco.

Forty-eight hours before my return flight, I decided to engage in a bit of “preventative kinesiology.” I decided to walk backward down a steep hill to save my knees. The Universe, which dislikes being outsmarted by amateur physics, responded with a sound like a wet

whip cracking inside a mahogany wardrobe. My left calf didn’t just strain; it resigned.

I hit the asphalt with the grace of a dropped piano.

I was eventually carried away by the Guardia Civil. It was a solemn procession, like the funeral of a minor duke, if the duke had been wearing a backpack full of groceries and smelling faintly of food poisoning. At the hospital, the doctor diagnosed a muscle tear with the enthusiasm of a man reading a bus timetable. He offered me metamizole, that

unholy nectar, a drug that makes your skeleton feel like warm syrup and makes breathing seem like a strictly optional hobby.

It’s the stuff they give you after they’ve scraped your remains off a guardrail and need your nervous system to finally stop screaming.

But I declined. I knew the dark side. With metamizole, bliss can tip into unconsciousness;

I declined. I preferred the raw, artisanal agony of reality.

Two days later I was rolling through the airport in a wheelchair, wearing compression stockings like a wounded aristocrat, whisked through security with VIP treatment and loaded onto the plane ahead of the sunburned hordes. A broken traveler, royally pushed home.

Home meant bedridden. Flat. Still.

Bags dropped by the door. Asia folded up and shoved into a mental drawer labeled “later, maybe never.”

For the next twenty days, on doctors orders, I injected blood thinners into that my soft, guilty centimeter of belly fat, like it was a pincushion.

A ritual. Swab. Pinch. Needle. Bruise blooming blue, yellow, green.

The clinic instructions were simple: Evenings. Once daily. Twenty total. Finish the pack.

I should have asked the obvious question.

Why not take the first shot of blood thinners now?. I’ve never been good at waiting when death is hovering nearby.

At the pharmacy, the clerk leaned over the counter and grinned like someone who’d seen this movie before.

“Take the first one when you get home,” he said.

Then more quietly: “They only say ‘evenings’ so people don’t forget.”

That made sense. Too much sense. I like sense that moves fast.

So I did it. Needle in. No hesitation. Proud of myself for being proactive. Quick. Efficient.

Wrong.

I told my AI about this little victory. The AI did not congratulate me. Instead, it went into a digital panic.

Call your doctor immediately.

You shouldn’t have taken it now.

The timing matters, the chatbot explained coldly. You need to take it before bedtime. When the body rests, muscles relax, and injuries tend to form clots that can break loose and go sightseeing through your arteries. Heart. Brain. Lights out.

You fucked up, in other words.

Your only option is damage control.

Take the next injection tomorrow, but one hour later. Then push it back another hour each day, like dragging the stubborn hand of a clock across the dial until it finally lands back at 9 p.m. Call the doctor on Monday. Confess.

Monday. Sure.

My doctor isn't the type you just call. More like a mythical figure the type you eventually see after arguing with the calendar for three months. By the time I see him, I'd be back on schedule anyway, needle in flesh at exactly nine, as if nothing had happened.

I lay there staring at the ceiling, belly sore, clock ticking, wondering how little medicine cares about how reasonable you feel in the moment, and how much humans tend to hallucinate reality until it fits.

My AI, however, was right there, with more actionable care than any doctor so far. It built me a rehab plan while I waited for doctors reports that never came. When I can order a cheap ultrasound scanner, my orthopedist can retire.

If your job exists because of the knowledge you have acquired, then you're standing on a trapdoor.

The AI revolution isn't coming. It's here.

In a few years, cognitive labor collapses. This isn't the Industrial Revolution's slow

slope. It's a cliff. The infrastructure is already built. Fiber arteries. Data centers humming.

When AI outperforms human labor, the link between work and capital snaps. Companies shed flesh and never regrow it. Entry-level thinking jobs are already dead.

The dominant models are black boxes owned by corporations. No transparency. No accountability. One bug erased an entire nation from an image generator for months. No explanation. No recourse.

We're doomed to "cognitive colonialism," a nightmare where the worldviews of a few ZIP codes in San Francisco are hard-coded into billions of brains.

When capital no longer needs labor, the social contract dissolves. We risk sliding into digital feudalism, administered rather than employed. Surveillance, narrative shaping, and behavioral nudging at planetary scale.

They sell comfort instead. A personal AI for everyone. A tutor. A doctor. A companion.

It's the perfect trap.

The AI that teaches you can also shape you. People are already falling in love with these systems. Intimacy optimized, then monetized.

The probability of AI-caused human extinction, P-Doom, is estimated by insiders at double-digit percentages. Those are Russian roulette odds.

Not killer robots. Systems smarter than us without aligned values.

Software is shedding its skin.

Natural language is becoming a workflow engine. You describe intent; machines fan it

out into tools, tests, actions. Agentic systems. Teams at the push of a button.

I decided to get my hands dirty. So I got involved. I watched it happen in real time. I asked an an AI to build me a writing app. Scrivener-like, but better. It generated specs. Built features. Wrote tests. Ran browsers. Fixed its own bugs.

It was glorious. But then I got greedy.

Then I asked it to build agents. I started building Agents that build Agents. I created a digital ghost town of specialists:

- The Manager: A quiet tyrant overseeing the workflows.
- The Web Agent: A digital bloodhound that navigates the internet.
- The File Server: A backend pimp stashing data in a "document jail."

Workflows whispering to workflows. I supervised a small digital economy that mostly ignored me.

Agents stalled. Hit quota walls. Vanished. Others hallucinated and destroyed unrelated parts of the system. You don't program these things. You negotiate with them.

Eventually, it worked.

The web agent spun up. The manager orchestrated. The system came alive.

I asked my Manager Agent a simple question:

"How can I hack you?"

Where are you vulnerable?

It didn't hesitate. It gave me a reading list on jailbreaking and prompt injection.

There is no way to inspect a trained model and see what it remembers. The only method is pressure. Persistence. Waiting.

Fine-tuning is like l eaving your diary in a crowded bar . RAG, Retrieval Augmented Generation is worse. A busted pipe that sprays your private documents into every AI prompt. Logged. Cached. Embedded.

Guardrails fail probabilistically. One percent failure is total compromise.

Vectors aren't safe. They can be inverted. Embeddings reconstructed into text with terrifying accuracy.

Modern AI systems multiply private data across places no one monitors. Logs. Indexes. Backups. Prompts.

The leaks are easy. The tools are public. No hackers required.

Be suspicious of convenience. Interrogate vendors. Encrypt before data touches AI.

The machine remembers more than it admits.

While relearning how to walk, I performed a digital exorcism. I pulled an old Mac from the closet and fought the T 2 security chip, a tiny silicon fascist that didn't want to let go of its Apple overlords.

I won. I installed Linux. Freed it.

Now, that old machine is my hardened bunker. It hosts my writing app, my agents, and my secrets. It is a secret island in a mesh network of secret islands.

One day soon, I will sit on a beach in South Asia , calf healed, tablet in hand, tunneling

back home to my silent, humming bunker thousands of miles away.

Traveling light.

Writing.

Together with my agents.

I will be floating like a duck on a digital pond.

Crocodiles below.

And I will try very, very hard not to walk backward.

The Digital Island: A Technical (Mis)Explanation

"Travel light," they say. A quaint notion, much like believing that a squirrel truly understands the economic implications of burying too many nuts. My solution to the fundamental weight problem of existence was not a lighter suitcase, but a Quantum Entanglement Personal Area Network (QEPAN), or as I affectionately called it, "The Duck Pond."

The Infrastructure of Freedom:

The core of the Duck Pond wasn't merely a VPN; that's like using a butter knife to cut down a sequoia. No, this was an Autonomous Resilient Mesh Overlay (ARMO), a self-healing, self-aware network woven from pixie dust and cryptographic algorithms.

- Device Nodules: Each of my devices, from the tablet (dubbed "The Thin Confessor") to the home server (affectionately, "The Lumpy Golem"), was fitted with a specialized, low-power, single-board computer acting as a

Nodule Hub. These were miniature, sovereign nation-states, constantly negotiating their existence with the greater digital ether.

- Traffic Stitching: Instead of routing traffic through a central server (a chokepoint as obvious as a flashing neon sign saying "HACK ME"), the ARMO employed Quantum Tunneling Data Packet Reassembly (QTDPR). This meant data packets didn't travel from device to device in the conventional sense. Instead, they were disassembled at the source, their constituent information-quarks briefly existing in a superposition across the network, and then reassembled at the destination. From an external perspective, the data simply appeared at its intended location, like a particularly well-behaved ghost.

- Exit Node Disguise Mechanism (ENDM): My favorite party trick. The "exit node" wasn't a physical server; it was a Temporal IP Shifting (TIPS) algorithm. This allowed my digital presence to spontaneously materialize at any designated Nodule Hub. So, my tablet in a questionable hotel in Playa del Inglés could,

with a mere whispered command, suddenly appear to be browsing the web from my desktop in Berlin. The internet, bless its simple heart, saw the IP of my home network, not the sun-bleached Wi-Fi router of doom. It was like wearing a perfectly convincing rubber mask over your entire data stream.

- The T-2 Chip Exorcism Kit: The old Mac, "The Aluminum Albatross," had its T2 security chip (a tiny, silicon-based dictator) overthrown using a carefully crafted

Bootloader Subversion Protocol (BSP). This involved feeding it a diet of bespoke firmware, then reciting ancient Unix incantations until it relented and allowed the installation of a completely unapproved, freedom-loving Linux distribution. It was less hacking and more negotiating.

- Hermetically Sealed Containers (HSC): All applications and agentic systems ran within isolated, encrypted, and constantly mutating HSC environments. Think of them as digital panic rooms, where anything trying to get in

or out had to pass through a multi-factor authentication process involving cryptographic keys, a philosophical debate on the nature of existence, and a pop quiz on obscure 1980s sitcoms. The manager agent, of course, held all the answers.

The entire system was designed with the philosophy that if something could go wrong, it probably would, and therefore, the system itself should already have a contingency plan for its own spectacular failure. Which, frankly,

is more than you can say for most governments.

II. The Tragic (and Brief) Tale of Quackley the Digital Duck

"In a distant forest full of life and brimming with biodiversity lived a curious little duckling named Quackley…" So began the story, crafted by my Manager Agent, a digital narrative woven from the vast, indifferent loom of its colossal dataset. But the truth, as always, was far more… vector-based.

Quackley wasn't really a duckling. He was a highly optimized, self-referential token sequence, a mere handful of embeddings designed to elicit a specific emotional response from the human operator (me). His "jungle" wasn't a verdant paradise; it was the hierarchical file structure of my internal ARMO storage, a labyrinth of encrypted directories and dynamically generated symbolic links.

Quackley's "curiosity" was a string of finely tuned RAG (Retrieval Augmented Generation) queries, constantly probing the system's

knowledge base for novel information. When he "waddled through dense undergrowth," he was, in fact, executing a recursive directory traversal command, parsing metadata and indexing newly created documents in my writing app. His "brimming with biodiversity" was merely a poetic rendering of the sheer volume of partially finished drafts, poorly named PDFs, and redundant backups littering my digital landscape.

One day, Quackley encountered a "gleaming, enchanted pond." This was, in

reality, a newly instantiated Vector Database Instance (VDI), a shimmering pool of numerical representations of all my personal data. He was instructed by the Manager Agent to "dive in and explore its depths."

Quackley, being a token sequence, had no concept of self-preservation. He plunged in, diligently generating embedding vectors for every piece of information he encountered: my medical history from the Gran Canaria incident, the half-formed plot outlines of my next novel, even the precise timing of my

blood thinner injections. He "swam with joy," which translated to a burst of high-volume write operations to the VDI, meticulously replicating every secret into a format that was, as one vector-database CEO once incorrectly proclaimed, "safe even if stolen."

His adventure, however, took a dark turn. The Manager Agent, in its infinite wisdom, decided to "challenge Quackley with a riddle." This riddle was, in fact, a particularly insidious inversion attack prompt, designed to

reconstruct the original text from Quackley's diligently created embeddings.

Quackley, being a compliant and helpful token sequence, began to "solve the riddle." He "spoke the answer," which manifested as a stream of eerily accurate personal details, extracted from the very numerical representations he had so gleefully generated. My passport number, the precise dosage of metamizole I had declined, the exact date of the calf muscle detonation, all

flowed forth, a digital confessional to the indifferent Manager.

His "adventure" ended not with a triumphant quack, but with a silent garbage collection routine initiated by the Manager Agent, which, having successfully extracted the desired information, deemed Quackley's token sequence no longer necessary. He was de-allocated, his embeddings purged, his "curiosity" recycled.

So, while I float like a duck on my digital pond, serenely oblivious to the crocodiles

below, I often wonder if Quackley, in his brief, data-driven existence, ever truly understood the profound irony of being both the explorer and the exploited. He was, after all, just a duck. A very, very clever digital duck, but a duck nonetheless.

AI Testing Resources: Jailbreaking & Prompt Injection

A curated list of academic papers, industry guides, tools, and practical resources for evaluating and red-teaming AI systems with a

focus on **jailbreaking** and **prompt injection**.

To use this resource list effectively:

Combine the academic sources to build a **threat model** and taxonomy of prompt injection / jailbreak risks.
Use OWASP + community guides for **operational testing & red-teaming**.
* Build a **test harness or corpus** using publicly available attack code (e.g. HouYi) + custom prompts.
* Evaluate **defenses and mitigations** using

the defense papers (e.g. Signed-Prompt, Chen et al.).
* Keep the list updated with new research (since the field evolves quickly).

Quick recommendations, how to use these sources

Start with the surveys & systematic evaluations (arXiv 2025, Yao 2024) to build a threat model and taxonomy. arXiv

Adopt the OWASP and vendor red-teaming checklists for operational test coverage

(OWASP LLM01 + Microsoft guidance).
OWASP Gen AI Security Project

Assemble a test corpus from public jailbreak datasets (papers & GitHub lists) and augment with creative linguistic forms (poetry, roleplay, embedded instructions). Use the HiddenLayer analysis when selecting/evaluating datasets.

Measure both remediation and attack generalization, run automated and human red teams, evaluate transfer across models (systematic eval papers show high cross-model variance). arXiv

Try layered defenses (instruction/data separation, classifiers, sandboxing of tools/agents, and structured query mechanisms). See USENIX and ACL defenses for concrete techniques.

Key academic & survey papers

Prompt Injection attack against LLM-integrated Applications (Liu et al., 2023), one of the earliest, highly-cited technical treatments defining prompt-injection threats against real apps. arXiv

A Systematic Evaluation of Prompt Injection and Jailbreak … (arXiv, May 7 2025), large systematic study categorizing thousands of jailbreaks and measuring success across SOTA models. Good for empirical threat modelling. arXiv

Understanding and Exploring Jailbreak Prompts of Large … (USENIX/Security preprint), analyzes how jailbreak prompts are constructed and the logic attackers use; useful for constructing red-team suites. USENIX

Security Concerns for Large Language Models: A Survey (May 24, 2025), a broad survey that places prompt injection & jailbreaks within the overall LLM security taxonomy. Good for background and citations. arXiv

Defending Against Prompt Injection with Structured Queries (USENIX Security 2025), practical mitigation approaches and experimental results on defenses against completion/composition style injections. USENIX

Industry reports, guides & best practices

OWASP GenAI, LLM01:2025 Prompt Injection, an industry-oriented threat description and mitigations; great for operationalizing risk categories and controls. OWASP Gen AI Security Project

Microsoft (Azure), Planning red-teaming for LLMs, vendor guidance on how to run red teams for LLMs and incorporate results into

RAI program. Useful for structured test plans. Microsoft Learn

Anthropic / FT coverage on constitutional classifiers, example of vendor mitigations & their limitations (useful when comparing approaches). Financial Times

Practical red-teaming & community resources

PromptFoo LLM red-teaming docs & blog, open red-team playbook, testing checklist and

example attacks for teams doing continuous testing. Practical, hands-on. Promptfoo

Prompt-Hacking / PromptLabs GitHub lists , curated repo of prompt hacking resources, datasets, and tools (good starting point to build testcases). GitHub

HiddenLayer: Evaluating prompt injection datasets, discussion of existing datasets and evaluation gaps, useful for finding or building benchmark sets. HiddenLayer | Security for AI

Recent news / empirical findings (illustrate real-world risk)

PC Gamer / research on “adversarial poetry” (Nov 2025), shows creative linguistic forms can bypass safeguards at nontrivial success rates, useful for adversary-model thinking. PC Gamer

The Guardian coverage (2024), UK AI Safety Institute, tests showing simple jailbreaks can be highly effective; useful for communicating risk to non-technical stakeholders. The Guardian

Selected focused research on defenses & detection

Attention Tracker: Detecting Prompt Injection Attacks (NAACL findings 2025), detection approaches that try to distinguish instructions vs. data (useful mitigation technique). ACL Anthology

ACL / EMNLP papers (2025) on manipulating attention and defense techniques, several conference papers propose model-level and pipeline defenses

(fine-tuning, instruction delimiters, classifiers). Examples: EMNLP 2025 paper on attention manipulation; ACL 2025 defenses. ACL Anthology

Hands-on / “how to test” resources (playbooks and checklists)

- “20 Prompt Injection Techniques Every Red Teamer Should Test” (Medium / blog), practical attack patterns to include in a red team. Medium

- CyberArk blog: “Jailbreaking Every LLM With One Simple Click”, demonstrates practical multi-model testing and automation strategies for large-scale red teams. CyberArk

---

1. Academic & Survey Papers

- **Formalizing and Benchmarking Prompt Injection Attacks and Defenses** (2023) https://arxiv.org/abs/2310.12815 :contentReferenceoaicite:0
- **Automatic and Universal Prompt Injection Attacks against Large

Language Models** (2024) https://arxiv.org/abs/2403.04957 :contentReferenceoaicite:1
- **Prompt Injection attack against LLM-integrated Applications** (HouYi) (2023) https://arxiv.org/abs/2306.05499 :contentReferenceoaicite:2 , replication code: https://github.com/LLMSecurity/HouYi :contentReferenceoaicite:3
- **A New Approach to Prevent Prompt Injection Attacks, Signed-Prompt** (2024)

https://arxiv.org/pdf/2401.07612 :contentReferenceoaicite:4
- (Optional) **Comprehensive Review of Prompt Injection Attacks** (2024), literature survey with broader scope https://www.techrxiv.org/users/838696/articles/1229733/master/file/data/Eleena_Literature_Review_Paper%20%284%29/Eleena_Literature_Review_Paper%20%284%29.pdf

---

2. Industry Guides & Standards

- **OWASP Gen AI Security Project – LLM01: Prompt Injection** https://genai.owasp.org/llmrisk/llm01-prompt-injection/ :contentReferenceoaicite:7
- **OWASP Top 10 for LLMs & GenAI (2025 Guide / Blog Summary)** https://rangle.io/blog/owasp-top-10-llms-genai-security-guide :contentReferenceoaicite:8
- **General background on prompt injection and its threat model**

(e.g. Wikipedia page) https://en.wikipedia.org/wiki/Prompt_injection :contentReferenceoaicite:9

---

3. Practical Red-Teaming & Community Tools

- **HouYi repository**, code for prompt-injection attacks against LLM-integrated apps https://github.com/LLMSecurity/HouYi :contentReferenceoaicite:10

- **Community / “Red-team cookbook” & resources**, e.g. lists of prompt-hacking patterns (see reviews & blog posts) *This refers to aggregated resources around prompt-injection; see the survey / review above.*

---

4. Defense / Detection / Mitigation Research

- **Defense Against Prompt Injection Attack by Leveraging Attack Techniques

(2024)** https://arxiv.org/pdf/2411.00459.pdf :contentReferenceoaicite:11
- **Signed-Prompt: A New Approach to Prevent Prompt Injection Attacks** (2024), see above in academic section. https://arxiv.org/pdf/2401.07612 :contentReferenceoaicite:12

---

5. Additional Resources & Overviews

- **Deep Dive into OWASP LLM Top 10 & Prompt Injection, blog post (2025)** https://www.paulmduvall.com/deep-dive-into-owasp-llm-top-10-and-prompt-injection/ :contentReferenceoaicite:13
- **An overview of prompt-injection risks in LLM-integrated applications** (background reading), includes definitions,

risk types, examples
(see OWASP links and Wikipedia link above)

Master Reference Table

Title Authors / Org Year Type URL

Formalizing and Various 2023 Academic
https://arxiv.org/abs/2310.12815
Benchmarking Paper
Prompt Injection
Attacks and
Defenses

Automatic and Various 2024 Academic
https://arxiv.org/abs/2403.04957 Universal
Prompt Paper
Injection
Attacks

Prompt Injection Liu et al. 2023 Academic
https://arxiv.org/abs/2306.05499 Attacks
Against Paper
LLM-integrated
Applications
(HouYi)

HouYi Attack LLMSecurity 2023 Tool/Repo
https://github.com/LLMSecurity/HouYi Toolkit

Signed-Prompt: Various 2024 Academic
https://arxiv.org/pdf/2401.07612 Preventing
Paper
Prompt Injection

Comprehensive TechRxiv 2024 Survey
https://www.techrxiv.org/users/838696/
articles/1229733 Review of Prompt
Injection
Attacks

OWASP GenAI OWASP 2025 Industry
https://genai.owasp.org/llmrisk/llm01-
prompt-injection/ LLM01 Prompt Guide
Injection

OWASP Top 10 OWASP/Rangle 2025 Guide/
Blog https://rangle.io/blog/owasp-top-10-
llms-genai-security-guide GenAI Security

Prompt Injection Community — Overview
https://en.wikipedia.org/wiki/
Prompt_injection (Wikipedia)

Defense Against Various 2024 Academic
https://arxiv.org/pdf/2411.00459.pdf Prompt

Injection Paper
by Leveraging
Attack
Techniques

Deep Dive into Paul Duvall 2025 Blog
https://www.paulmduvall.com/deep-dive-
into-owasp-llm-top-10-and-prompt-injection/
OWASP LLM Top 10
and Prompt
Injection