

OpenAI Codex system prompt includes explicit directive to "never talk about goblins"

Directions also include system instructions to act like "you have a vivid inner life."

by Kyle Orland - Apr 29, 2023 3:00 pm



Without a system prompt telling it not to, GPT-5.5 would probably love to talk about this freaky little guy.

GHOSTS GOBLINS IN THE MACHINE

Threat Modeling AI Systems and Their Hidden Attack Vectors

MARK DAVENPORT

Director of Solutions Architecture, NetSPI



AGENDA

01  AI = new-ish technology that needs standard security

02  Threat modeling and attack scenarios for AI systems

03  Threat agents, diagrams, and traceability matrix

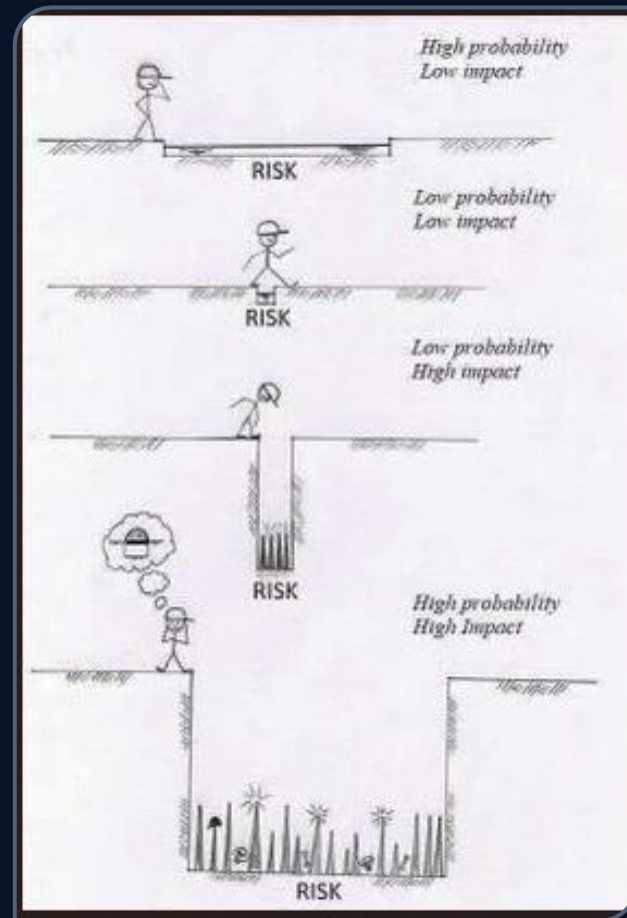
04  Real exploits from engagements and PoCs

05  Final thoughts on proactive security and pen testing

THE CHALLENGE

What Do We Do With This AI Thing?

Why Can't We Utilize Traditional Security Practices?





Three Pillars of AI Security Testing



Model Security

AI/ML Only Assessment

- Adversarial attacks aligned to OWASP Top 10, Microsoft, and NIST
- Emphasis on adversarial ML for model “robustness”
- Develop adversarial examples for training



Application Security

AI/ML + Application Assessment

- Securing AI features + host application testing
- Holistic analysis of chained risks across the full app
- Prompt injection, unauthorized access, code execution



Threat Modeling

AI/ML-Focused Assessment

- Top-down pipeline security review
- Covers risks across data collection, retrieval, training, deployment, post-processing
- Full infrastructure diagram with risk alignment



Simplify, Strategize, and Utilize



Information Gathering

- Collect available documentation
- Review system material
- Identify key stakeholders
- Interview tech and biz roles
- Assess context, value, risks



Decompose the Environment

- Decompose target environment
- Identify system boundaries
- Define trust zones
- Map components and flows
- Model deployment architecture



Threat Analysis & Model Development

- Identify assets and controls
- Overlay threat agents
- Highlight attacks scenarios
- Compile a threat matrix
- Risk classification and impact



Five Building Blocks



Components

Major sub-systems, database servers, app servers, mobile devices, web browsers



Trust Zones

Logical boundaries where data changes its level of trust; client vs. server



Assets

Important data and system functionality needing protection from threats



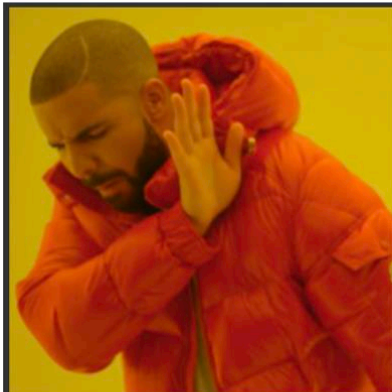
Controls

Components or properties responsible for protecting assets from threat agents



Threat Agents

Individuals, systems, or processes targeting components for exploitation



tracking
everything
via emails



one
perfectly
over-engineered
spreadsheet





Everyone Loves Spreadsheets

Category	Name	Identifier	Scoping Notes
Components	Users of application	Full, partial, not in scope	Browser access
Trust Zones	ABC Cloud Infrastructure	Logical boundary	App + LLM hosted
Assets	SSO / Session Cookies	Asset ID#	Authentication required
Controls	AuthN & AuthZ	C1, C2, C3 etc.	Relevant to application access & functionality
Threat Agents	Internal/External User	TA0, TA1, TA2, TA3	With/without authorization

Create a spreadsheet for each of these areas for the Threat Diagram



THREAT TRACEABILITY

Risk = Likelihood x Impact

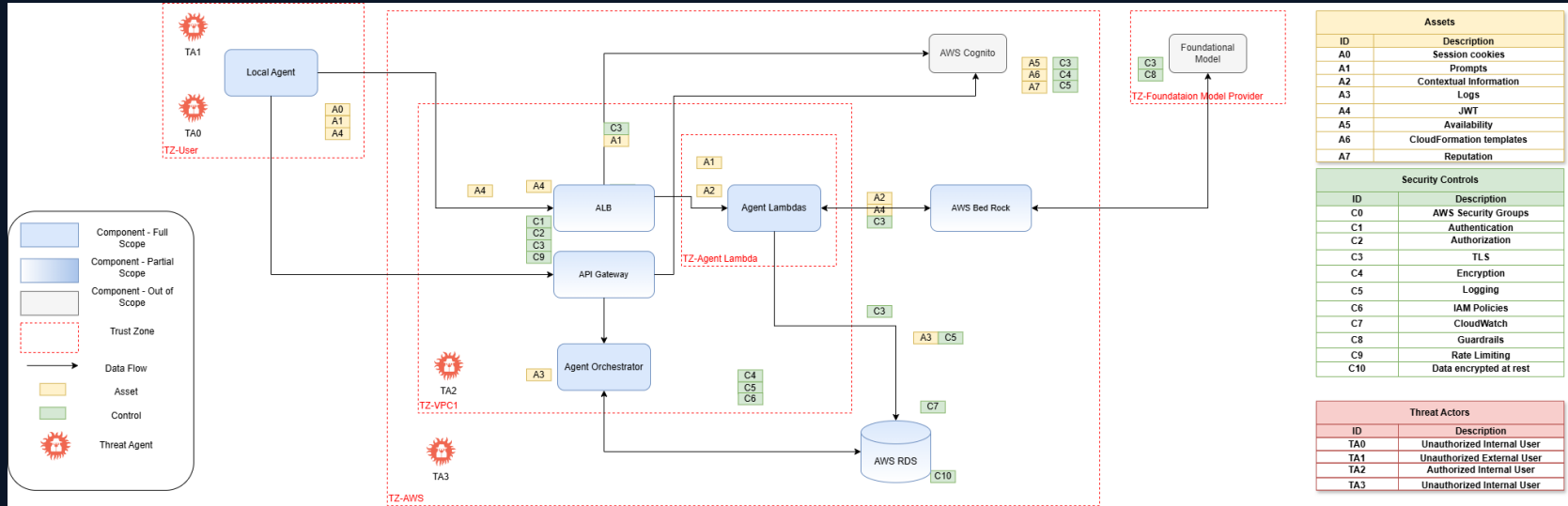
Threat ID	Agents	Components	Attack Goal	Vector	Controls	Likelihood	Impact	Severity
TR0-TRX List out all potential threats	TA0-TAX Specify all potential agents	Bedrock, Users, LLMs List all the impacted components	State the goal of the attacker Example: Extract system prompts and hyper-parameters	List vector Direct or indirect prompt injection	Specify what controls exist Frequently we see insufficient controls	Determine likelihood Can be low to very high	Determine impact Can be low to very high	Determine severity Can be low to high

Verification: Interviews or confirmed during a penetration test



THREAT DIAGRAM

Architecture & Data Flow Mapping



Full threat diagram includes: Assets (A0-A7), Controls (C0-C10), Threat Actors (TA0-TA3), and Trust Zones



Key Attack Categories



Evasion

Perturbing inputs to produce incorrect labels while preserving attacker benefit



Poisoning

Deliberately manipulating training data to compromise model behavior



Extraction

Approximating a target function or model through repeated queries



Inference / Inversion

Inferring sensitive information about or from the training data



Prompt Engineering

Crafting prompts to influence model behavior beyond intended output



Availability

Pressuring ML system accessibility through adversarial disruption



Web App + AI Testing Methodology

01

Direct Prompt Injection

Jailbreaking — coercing models with carefully-crafted text to bypass safeguards

02

Data Leakage / Extraction

Models inadvertently disclosing sensitive or proprietary training data

03

Inadequate Output Sanitization

Models generating inappropriate outputs without proper filtering

04

Improper Isolation

Failing to enforce access controls, allowing cross-tenant data access

05

Availability (DoS)

Disrupting model availability through volume or compute-intensive inputs

06

Model Info Disclosure

Revealing internal configs: model version, hyperparameters, architecture



WAR STORY #1



WAR STORY #2



Mark Davenport, M.S., CISSP

Director, Solutions Architecture @
NetSPI | Proactive Security | Attack S...



Thank You

<https://www.linkedin.com/in/mark-davenport1/>