PENTESTERRA

AI PENETRATION TESTING

AI Penetration Testing That Runs Real Tools, Not Just an LLM

PentestBrain is Pentesterra's autonomous exploitation engine - a ReAct AI loop that verifies CVEs with real tools, chains findings into real attack paths, and proves whether your remediation actually closed the risk. We are not an LLM wrapper.

ReAct

Reasoning + real tool execution loop

Real Tools

tcp_probe, http_probe, nuclei_scan, msf_exploit

Proof

Every finding backed by real evidence

WE ARE NOT AN LLM WRAPPER

DIMENSION	TYPICAL "AI PENTEST" TOOL	PENTESTBRAIN
How it verifies findings	Asks an LLM to guess if it's exploitable	Runs real tools and reads real output
Attack paths	Summarizes a list of CVEs	Builds a real cross-domain attack chain
Evidence	A confidence score with no proof	Request/response evidence per finding
Remediation check	Not tracked	Re-verifies the fix on the next cycle

HOW PENTESTBRAIN WORKS

ReAct Loop With Real Tools

PentestBrain doesn't guess. It runs a reasoning-and-acting loop that calls real tools and reads their actual output before deciding the next step.

- → Every step backed by a real tool call, not an LLM guess
- → CVE verification against real service responses
- → Adapts the plan based on what each tool actually returns
- → Full step-by-step log, auditable end to end

Attack Chains, Not Isolated Findings

Attack chains trace paths from initial access to lateral movement and business impact.

- → Cross-domain graph: web + network + DevGuard findings
- → MITRE ATT&CK phase mapping per chain node
- → Shows how individual findings compound into real risk
- → Updated on every scan cycle, not just once

Safe Exploitation Boundaries

Autonomous doesn't mean unbounded.

→ Configurable rulesets per engagement scope

→ Non-destructive validation by default

→ Human review gate before any high-impact action

→ Consistent, reproducible verification across cycles

Proof, Not Just a Score

Every verified finding comes with the evidence PentestBrain collected to prove it.

→ Real request/response evidence per finding

→ Re-verification confirms remediation actually worked

→ Regression detection if a fixed issue reopens

→ Compliance-ready evidence trail per cycle

UNDER THE HOOD

A Real Tool Arsenal, Not a Prompt

Each step in the loop dispatches to a real tool and waits for its actual output before deciding what to do next.

- → tcp_probe / http_probe - live connection and banner checks against the real service
- → nuclei_scan - template-based vulnerability verification
- → msf_exploit - runs an actual Metasploit resource script (msfconsole), not a simulated exploit
- → credential_test - live authentication attempts against the target

Anthropic First, With a Fallback

The reasoning model itself is chosen for reliability on this specific job.

- → Claude as the primary reasoning model - strong tool use, doesn't refuse PoC generation for known CVEs
- → OpenAI as an automatic fallback if no Anthropic key is configured
- → Every LLM decision is one step in an auditable ReAct loop, not a black box

Watch It Think, Live

Exploitation isn't a black box you wait on.

- → Real-time SSE thought stream in the UI - see each decision as it happens
- → Full step-by-step reasoning log stored per session, auditable after the fact
- → An artifact from one step (a stolen cookie, a valid credential) is explicitly passed to the next

Strict Separation of Reasoning and Execution

The AI brain never touches your network directly.

- → Reasoning runs on the Worker node - it never runs nuclei, nmap, or msfconsole itself
- → All network operations are dispatched to and executed by Scanner nodes only
- → Every exploitation attempt is logged and attributable, end to end

FAQ

Does PentestBrain actually run exploits, or just simulate them?

It runs them for real. msf_exploit dispatches an actual msfconsole resource script against the target; nuclei_scan and the network probes make real connections and read real responses. Nothing is simulated or guessed by the LLM.

What happens if the reasoning model gets it wrong?

Every action stays inside configurable exploitation boundaries and is non-destructive by default. A human-in-the-loop review gate can be required before any high-impact action, and the full step-by-step reasoning log is stored and auditable after the fact.

Can I watch it work instead of waiting for a final report?

Yes. The session streams a real-time thought log over SSE in the UI, so you see each tool call and decision as PentestBrain makes it, not just the final verdict.

Does it prove a fix actually worked?

Yes. On the next cycle, PentestBrain re-verifies previously exploited findings and raises a regression alert if something that was fixed has reopened - your remediation gets checked, not just your original finding.

See PentestBrain verify a real CVE

Autonomous exploitation with proof, not a confidence score.

Request a Demo

info@pentesterra.com · pentesterra.com/ai-penetration-testing