

ROOTSHELL PLATFORM · MCP SERVER · RESEARCH

What an LLM Adds to Exposure Management

We connected an LLM to the Rootshell Platform through our new MCP Server and asked it to work across a test estate. Here is what it could do, and where the line sits.

Tony Allen · September 2026

We recently released the Rootshell Platform MCP Server into beta. It lets organisations connect their own AI to their exposure data. Before putting it in clients' hands, we wanted to see what an LLM could usefully do with the information the server provides today.

We connected one to a test environment: thousands of findings across nearly 200 assets, including endpoints, cloud servers, network devices and penetration test results. Everything in it, and in the visuals below, is test data. We also gave it a strict rulebook. It could read records, but it could not change them, scan an asset or try to exploit anything. It had to show the evidence behind each conclusion and say plainly when a link was unproven.

The result was a useful demonstration of how AI can complement the platform and our testers. The platform brings findings, exploit intelligence, asset context, remediation and reporting together. An LLM can use that foundation, combine it with information a client chooses to provide, and ask what the records might mean together.

Start with the platform

The Rootshell Platform already does a great deal of the work needed to turn security data into action. It brings scanner results, penetration tests and assessments into one place. It tracks recurring findings across scans and prioritises issues using severity, asset priority and client-defined tags. Clients can manage remediation through automation, integrations and live dashboards, including SLA and time-to-fix reporting.

Daily Exploit Detection marks the findings that matter most at a glance:



Black bug. A known working exploit exists for this vulnerability.



Red bug. The vulnerability is being actively exploited in the wild right now.

That matters because an LLM needs a reliable starting point. It cannot compensate for missing evidence by sounding confident. Its value comes from working across the evidence the platform holds and the additional context a client has about their own environment.

The platform tells you which findings are dangerous. An LLM helps you see where they connect, what is missing around them, and what that means for each person reading.

The attack path that caught our attention

The most interesting output was a possible route from a lower-priority endpoint towards a high-priority cloud production account. The endpoint had a red-bug browser flaw listed in CISA's Known Exploited Vulnerabilities catalogue. The platform identified the finding; the LLM asked what that endpoint might lead to.

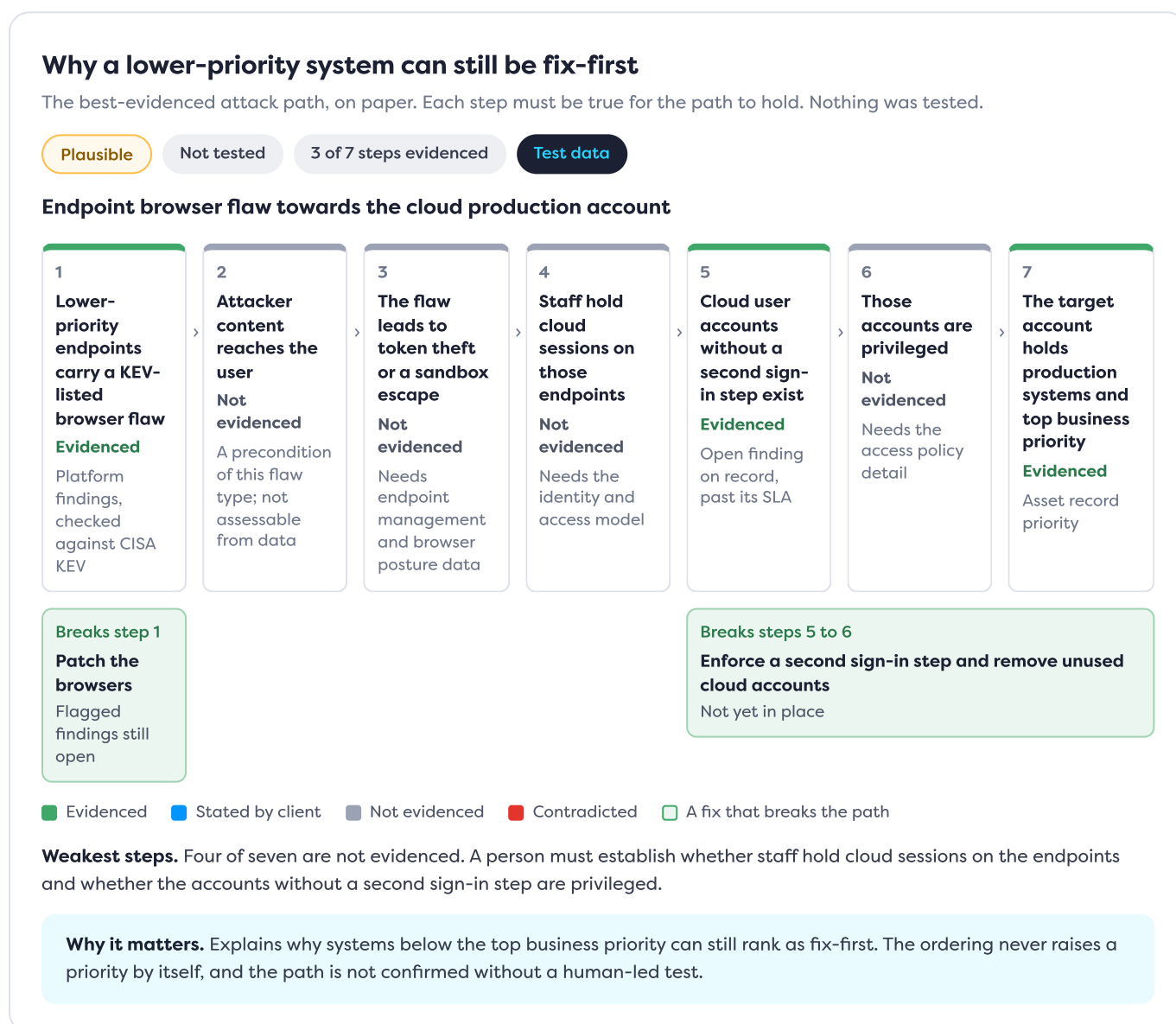


Figure 1. A view the LLM built for itself from platform records. Test data. Every step is graded by the evidence behind it, and the two fixes show where the proposed route could be broken.

It laid the route out as seven steps, and built the view below itself from platform records. Three steps were supported by records: the endpoint finding, cloud user accounts without a second sign-in step, and the production account's high business priority. Four steps were not evidenced. We did not know from the available records whether attacker-controlled content would reach that user, whether the browser flaw could lead beyond its sandbox on that endpoint, whether staff used the cloud account from that device, or whether the accounts without the second sign-in step were privileged.

The LLM labelled the path plausible, not tested. That distinction is the point. It did not discover a working attack or prove that the laptop was a route into production. It gave our testers a set of precise questions: who uses the endpoint, what browser and controls are present, which accounts are accessed from it, and what privileges and sign-in protections those accounts have.

It also showed where a fix could interrupt the proposed route. Patching the browser addresses the first evidenced step. Enforcing the second sign-in step and removing unused cloud accounts addresses a later one. Those are sensible actions to consider in their own right; the diagram made their relationship easier to see without claiming the whole route was real.

We liked this because it gave a reason to investigate an issue that could otherwise look less urgent when viewed only through the endpoint's business priority. It was a hypothesis with its evidence and gaps exposed, ready for a human to test. It did not change the client's priority rating or automatically reorder the fix list.

Joining records to the client's own context

We then provided an export of cloud servers and firewall rules. The LLM compared it with the platform's asset records and scan history. It highlighted internet-facing servers whose firewall rules allowed remote access or web traffic from the internet, despite recent scans finding nothing significant. That is a useful question for a tester: were the scans in scope for those services, and do the records describe the same exposure? Open ports alone do not establish a vulnerability, and a scan without a significant finding does not establish that every reachable service is safe.

The comparison also surfaced a server absent from the asset register and duplicate gateway records. Our team checked those results and identified the setting responsible. The useful part was the cross-check: the platform had one set of evidence, the client-provided export had another, and the LLM helped us compare them quickly.

The same principle applies to priority. One server was recorded as internal, but the cloud export showed a public address on a matching host. Once our team confirmed they were the same machine, the LLM recommended moving its finding to the top of the fix list. The asset's business priority was already right; what changed was what it was exposed to. Elsewhere, it flagged a laptop carrying a red-bug finding whose asset record had no name, owner or priority at all. In each case the client and our team decided whether the record or the priority should change.

Looking beyond the platform

The platform's black and red bugs say that an exploit exists or is being used. The LLM then went looking for the wider story, using public sources it could cite. It checked each flagged flaw against CISA's Known Exploited Vulnerabilities catalogue, and read public threat reporting for any group named as using the flaws. Where a security vendor had named the actors, the LLM put those names on the page, with the source and date beside each one.

The exploit story: from systems to attackers

Follow the flagged exploits from the systems they sit on, to how widely they are exploited, who is reported to use them and how recently.

Test data

Systems and counts are test data

Flaws, dates and actors are public

1 Which systems carry a flagged exploit?

Systems affected by an issue the platform flags with a black or red bug, grouped by the priority the client gave the system. Priority 1 is highest.

Priority 1	0 of 40
Priority 2	2 of 60
Priority 4 and 5	6 of 45
No priority set	2 of 30

Most flagged exploits sit on lower-priority or unrated systems, which is why the attack path view matters.

2 How widely is each flaw exploited?

KEV means CISA lists the flaw as known to be exploited. Flaws and dates are public; system counts are test data.

Chromium V8 type confusion	3 systems	KEV 4 Sep
CVE-2026-85046 · Chrome · priority 4, 5 and unrated · CISA due date passed		
Chromium V8 out-of-bounds write	3 systems	KEV 9 Sep
CVE-2026-87491 · Chrome and Edge · priority 4 and 5		
Windows heap-based buffer overflow	1 system	KEV 8 Sep
CVE-2026-85880 · Windows · priority 4		
Windows link following	1 system	KEV 8 Sep
CVE-2026-81963 · Windows · priority 4		
Skia out-of-bounds write	1 system	KEV 13 Mar
CVE-2026-3909 · Chrome and Edge · priority 2		

3 Who is reported to be exploiting them?

Names come from public reports. Context only: it never changes a rank.

UTA0560 Volexity, 9 Sep 2026

Reported targeting non-governmental organisations. Named for both V8 flaws and the Windows heap overflow.

JungleBamboo Volexity, 9 Sep 2026

Also tracked as APT31, Violet Typhoon and TA412. Named for the same three flaws; no target sector given by this source.

No source read names the test client's sector as a target

Two flagged flaws have no published attribution in the sources read. Not evidenced does not mean not targeted.

4 How recent is this?

Dated events from public sources and the platform, oldest first.

13 Mar	CISA added the Skia flaw to KEV
4 Sep	CISA added the V8 type confusion flaw to KEV
8 Sep	CISA added both Windows flaws to KEV
9 Sep	CISA added the V8 out-of-bounds flaw to KEV; Volexity names the actors
Run -1	Test estate: new red-bug findings, including on an endpoint with no owner recorded
Run day	Test estate: same red-bug set persists

Figure 2. The LLM followed each flagged exploit out of the platform to public sources and back. Systems and counts are test data; flaw names, KEV dates and threat-actor attribution come from public sources. Attribution is context and never changes a priority.

Two rules kept this honest. Every item carries its source, so a reader can check it. And attribution stays context: knowing which group is reported to use a flaw adds colour for a board, but it never changes a priority, and the LLM never implies a client is being targeted unless a source says so.

One analysis, different questions

A tester wants to know what to verify next and why. A remediation owner wants to know which action breaks a possible route. A board wants a short account of exposure and progress. An LLM can explain the same underlying evidence for each audience and answer follow-up questions in plain English.

A CISO asks something sharper: which threat actors are exploiting the flaws on our systems today, and are they going after our sector? That is threat intelligence that hits home. Because the LLM starts from the client's own open findings, it can answer from public reporting: these flaws, on these systems, named against these actors by this source, on this date. It also says plainly when no source names the client's sector as a target. That keeps the answer relevant without inflating it, and it never changes a priority on its own.

That does not replace the platform's live dashboards, workflows or metrics. It makes it easier to interrogate them alongside other information and to turn a specific question into a focused worklist.

The Rootshell Platform provides	An LLM connected through the MCP Server can add
Consolidated findings and daily exploit intelligence	Questions about how findings may relate across assets and accounts
Client-defined asset priorities, owners and tags	Checks where other evidence suggests context should be reviewed
Finding history, remediation workflows and dashboards	Explanations and worklists tailored to the person asking
A continuing record of what has been found and fixed	Comparisons with cloud configuration or other data the client supplies

The boundary matters

It is now technically possible to give an LLM tools that scan, probe and try to exploit live systems on its own. The question is whether that is wise today. Our answer is no. For live client assets, human-led is the safest approach, and it is the one our clients choose.

That comes from how testing works in practice. When our testers find something that could be taken further, they stop and ask the client: do you want us to go further? Clients rarely say yes to actively exploiting a live production system. They weigh the value of proof against the risk of an outage, data exposure or a disrupted business process, and they usually prefer to fix the issue on the evidence already in hand. That decision belongs to the client and to an accountable tester, not to a model acting on its own.

An LLM can suggest where to look. A person decides whether to touch a live system, and takes responsibility for it.

There is a second reason. LLMs are not always right, but they can sound remarkably convincing when they are wrong. We saw it in this test. Reading through the MCP Server, the LLM sometimes concluded that finding history or SLA measures were missing, when those capabilities already existed elsewhere in the platform. It stated those conclusions fluently and with confidence. Our team caught them because they know the platform. A fluent answer is not evidence, and an autonomous tool acting on a confident mistake against a live system is a risk no client should carry.

The same applies to the data. The MCP Server is a read-only window, and the LLM sees only what it returns, which may be less than the platform itself knows. If you connect an LLM to your own data, don't assume it is reading everything.

So we kept a person in the loop at every step that mattered:

- **The LLM reads, correlates and proposes.** Paths, gaps, questions and recommended priority changes, each with its evidence and its unknowns.
- **People check.** Our team verified the AI's conclusions against the underlying records before anyone acted on them.
- **People decide.** The client and our testers decided whether a record or a priority should change.
- **People test.** Where a question could only be settled by touching a live system, the LLM stopped and handed it to an authorised tester, working within the client's permission.

This experiment was separate from how we already use AI in testing. Outside the MCP Server, our continuous testing service uses AI tooling as part of the offensive security process itself, in penetration tests and ongoing assessments. The same principle applies there: our testers direct the tools, review what they produce and work within the scope and authorisation the client has agreed. We describe that approach in more detail in *Human-Led AI Testing: Why AI Won't Replace Penetration Testers...Yet* on our blog.

That is human-led, AI-augmented continuous testing. AI takes on the reading and joining up that no team has time to do by hand, every day. Our testers bring judgement, accountability and the client's trust. Rootshell maintains the evidence and the remediation process. The final judgement stays with people who understand the environment.

As we develop the MCP Server, we are exposing more finding history and asset relationships, so clients spend less time working out where to look and more time reducing the exposure that matters.

Bring your own AI to your exposure data

The Rootshell Platform MCP Server is in beta and coming to clients soon. To see what an LLM could find in your own data, talk to our team at rootshellsecurity.net.

Method: data analysis of a test environment only. No live checks, probing or exploitation were performed. All systems, figures and counts are test data. Public vulnerability identifiers, CISA KEV dates and threat-actor attribution are taken from their public sources (CISA, NVD and Volexity).