SYNACK SECURITY RESEARCH REPORT

NatJack

When 30 Years of Network Design Becomes the Vulnerability

How a legacy trust assumption in network address translation exposes nearly every router, container, and cloud platform on the internet.

RESEARCH CREDIT

SODIUM-24, LLC • Malcolm Stagg • Synack Red Team

7.2–9.6	2
CVSS	CVEs

EXECUTIVE SUMMARY

Network address translation was never built to be a security boundary. It was built to solve a shortage of IPv4 addresses, which was a plumbing decision made under the assumption that everyone behind the same NAT table was trustworthy. That assumption held for decades, because for most of the internet's history, it was true.

It isn't anymore. **NatJack** is a newly disclosed attack class that manipulates the NAT table itself. It hijacks active TCP connections, poisons UDP DNS responses, and forces denial of service against any device performing network address translation. That includes consumer routers, enterprise firewalls, cloud NAT gateways, and the virtual bridges inside Docker, Kubernetes, and Hyper-V. Independent testing found that all evaluated NAT implementations were vulnerable to some or all of the NatJack techniques.

No single patch fixes this because NatJack is a design assumption shared across an entire category of infrastructure. It also serves as a case study in vulnerabilities that automated tools miss.

All tested NAT devices vulnerable

High potential for real-world exploitation

CVSS 7.2 – 9.6

How it was found. Synack Red Team researcher Malcolm Stagg found NatJack by accident, during a Synack engagement testing a customer's virtual machine. Flooding a NAT device with sequenced packets revealed that replies sometimes came back mismatched, which was a sign the NAT table itself could be manipulated. Tracing that anomaly through the router's DD-WRT firmware led to Netfilter, the connection-tracking library behind most Linux-based NAT. A related discovery, drawn from RFC 1337's TCP TIME-WAIT behavior, cut the attack's execution time from seconds to near-instant. This was enough to hijack a live HTTP connection to a public server before the request completed.

01 | TECHNICAL OVERVIEW

NatJack is a newly identified network address translation (NAT) table manipulation attack class effective against most virtual and physical network infrastructure performing NAT. These attacks can be used to:

Hijack TCP connections

Take over an existing TCP connection traversing the NAT.

Poison DNS responses

Intercept and maliciously alter UDP DNS responses.

Force denial of service

Exhaust the NAT table to break connectivity.

Identify connection ports

Determine which port a NAT assigned to an active connection.

ROOT CAUSE: Cooperative Design in an Adversarial World

The root cause of NatJack includes legacy assumptions about sequencing, port allocation, and trust between devices sharing a table. Two CVEs have been assigned, with vendor discussions ongoing and more CVEs likely.

CVE-2026-56181

Microsoft Windows NAT affecting Hyper-V in a downstream spoofing configuration.

CVE-2026-63913

Linux netfilter conntrack affecting the connection-tracking subsystem behind Linux NAT.

02 | WHY THIS CLASS OF VULNERABILITY EVADES CONVENTIONAL TESTING

A scanner looks for known signatures like a CVE, a version string, a misconfiguration pattern someone has already catalogued. NatJack isn't any of those things. It's a behavior that emerges from how NAT tables have always worked.

For anyone relying on point-in-time or purely automated testing, you cannot fuzz your way to a discovery like this. It surfaced through manual, adversarial testing of how a NAT device behaves under load, tracing an anomaly in reply sequencing back through the router's firmware to the underlying NAT library. The same underlying flaw was independently confirmed across Windows, Linux, macOS, and other operating systems that share no common NAT codebase. It's a shared design assumption, not an isolated bug.

Bottom line: Design-level flaws surface only when someone tests the assumptions a system was built on.

03 | SCOPE & IMPACT

All tested routers and network infrastructure devices performing any form of NAT were found vulnerable to some or all NatJack techniques. That includes containerization platforms and hypervisors, affected through their virtual bridges and switches, along with many public cloud container and virtualization services.


Routers & firewalls


Docker & Kubernetes


Hypervisors (Hyper-V)


Public cloud services

Advisory at a glance

CVSS	7.2 – 9.6 (High to Critical)
Disclosed	May 1, 2026
Exploitation	High potential for real-world exploitation
Affected products	Most vendors and products using NAT
Attack vector	Network, unauthenticated
Patch availability	Partial (see mitigation guidance)

04 | DETECTION & MITIGATION GUIDANCE

Potential Detection Signals

A full or near-full NAT table; a flood of TCP/UDP packets from a downstream endpoint across a wide port range; the same IP at two physical addresses (or one address using multiple IPs); anomalous SYN/RST sequences; a flood of RST packets with varying sequence numbers; sequence numbers far outside the current TCP window; unusually low TTLs.

Mitigation by category

Encryption	Use TLS (including HTTPS) everywhere, even internally because it prevents useful connection hijacking outright. Use DNSSEC, plus DNS-over-HTTPS or DNS-over-TLS, to stop DNS response spoofing.
Containers & virtualization	Disable network access for untrusted containers/VMs where possible, or isolate them on a private network. Run untrusted workloads on their own host. Avoid root. Drop default permissions (--cap-drop=NET_RAW).
Kubernetes	Separate untrusted and trusted pods by node. Avoid root. Drop default capabilities via securityContext.capabilities.drop.
Cloud	Don't place untrusted and trusted workloads behind the same NAT or internet gateway. Use dedicated IPs for serverless workloads rather than shared ones.
Routers & firewalls	Enable IP source protection (e.g., IP Source Guard). Segment untrusted users into a separate subnet/VLAN with ACLs. Cap TCP/UDP connections per client (under ~10k). Disable loose conntrack modes, port preservation, and endpoint-independent mapping where possible. Enable ISN randomization.

Available patches include a Linux kernel patch (kernel 6.6.142 and higher, which increases attack complexity but is not a complete fix) and a FreeBSD incidental patch (15.0 and higher, which reduces the practicality of most attacks). Full remediation will be an ongoing, multi-vendor effort. Track updates at natjack.io.

05 | FREQUENTLY ASKED QUESTIONS

Is my router vulnerable?

Almost certainly, if it performs NAT, which nearly all consumer and business routers do. But vulnerable doesn't mean at-risk: home users are generally safe, because an attack requires an untrusted, privileged user already positioned behind the router. Businesses and service providers face real risk depending on network architecture.

Does this affect cloud and containers?

Yes. Docker, Kubernetes, and hypervisors like Hyper-V are affected through their virtual bridges and switches, and many public cloud container and virtualization services inherit the same exposure.

Is there a complete fix?

Not yet. Available patches raise attack complexity but do not fully close the class. Expect incremental, vendor-by-vendor remediation over an extended period.

Vulnerable is not the same as at-risk. Prioritize remediation where untrusted, privileged workloads share a NAT with trusted ones.

ABOUT THE RESEARCH

NatJack was discovered by **Malcolm Stagg**, an independent researcher for **SODIUM-24, LLC** who is also a security researcher on the **Synack Red Team**.

Full technical details were presented at **Black Hat 2026** — *Breaking Trust Boundaries: Exploiting Design Assumptions in Network Infrastructure*.

ABOUT SYNACK

Synack is the AI + human penetration testing platform for continuous security validation. Sara AI Pentesting combines agentic AI with the Synack Red Team, a rigorously vetted community of security researchers, to expand testing coverage and prove real-world exploitability. The Synack Platform gives organizations control and visibility across testing activity, validated findings and remediation status. Synack supports point-in-time and continuous penetration testing across web applications, APIs, mobile applications, cloud and host infrastructure, internal environments and AI or LLM systems. Founded by former NSA operatives, Synack has enabled nearly 10 million hours of security testing to protect critical assets across enterprise, public-sector and highly regulated environments. Learn more at synack.com and on LinkedIn.