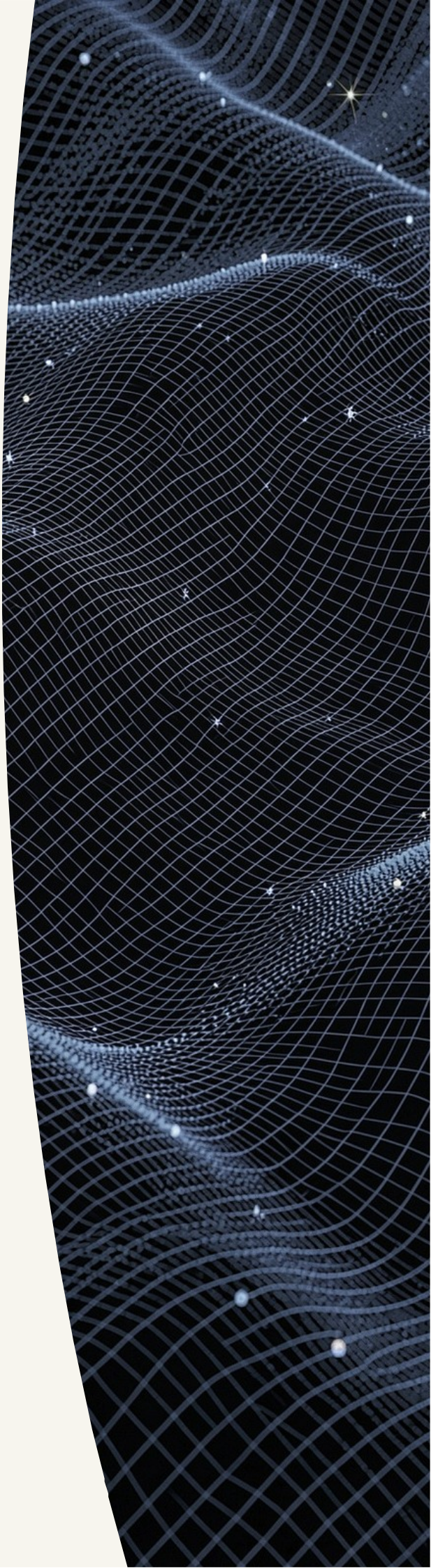
DDoS Attacks: How Websites Get Flooded, and How They Fight Back

A technical deep-dive into distributed denial-of-service attacks — how they work, why ordinary defenses fail, how the world's largest networks fight back, and what India's own infrastructure looks like from the inside.

NETWORK SECURITY INFRASTRUCTURE DDOS MITIGATION



Packets, Botnets, and the Flood

How the Internet Moves Data

The internet breaks every page into tiny **packets**, each carrying data and a label showing where it needs to go — an **IP address** unique to every device. Every request, every page load, every video stream is just packets traveling between addresses.

The Botnet Mechanism

Someone secretly takes control of thousands of ordinary computers and phones, without their owners finding out, forming a **botnet**. Whoever controls it can give one command, and every device obeys at once — all hitting the same website at the same moment.

What a DDoS Attack Actually Is

A server might handle a few hundred visitors at a time. If fifty thousand show up together, real people can't get through, and the site goes down — not because anyone broke it, but because it got overwhelmed on purpose.

A **DDoS attack** — Distributed Denial of Service — is a crowd blocking the entrance so nobody else can get in. The attack is distributed because it comes from many devices at once, making it far harder to block than a single source.

Why "Distributed" Matters

A single attacker from one IP address is easy to block. Fifty thousand infected devices spread across dozens of countries, each sending legitimate-looking packets, is an entirely different problem — one that requires infrastructure-scale solutions.

Why an Ordinary Firewall Can't Stop It

What a Firewall Does

Every network has a **firewall**, working like a security guard checking who's coming in and limiting how many can enter at once. For everyday problems — port scans, known malicious IPs, suspicious patterns — this works fine.

The Bandwidth Ceiling

But it has a limit it can't get past: every connection can only carry a certain amount of data per second, its **bandwidth**. Home connections are far narrower than data centre ones. If an attack sends more traffic than even a wide connection can carry, the connection fills up before the firewall can check a single packet.

> No software fixes a connection that's already full. This is the fundamental constraint that makes volumetric DDoS attacks so difficult — the problem exists at the physical layer, before any inspection logic can run.

The Road Jam Analogy

Think of it like a road so jammed the guard never sees the vehicles arrive. The firewall is still running, still checking rules — but the pipe feeding it is completely saturated. Every packet, legitimate or malicious, is simply dropped at the edge. The site goes dark not because the firewall failed, but because the connection it sits behind was overwhelmed first.

Normal Checks

Bandwidth Saturation

One Address, Many Machines: The IPv4 Squeeze

The Address Exhaustion Problem

Every device joining an attack, and every server defending against one, needs an address to be found by. The system the internet still mostly runs on, **IPv4**, only has about **4.3 billion** possible addresses. The central pool of unassigned IPv4 addresses ran dry in **February 2011**, and regional registries exhausted their own remaining stock region by region through the following several years.

Carrier-Grade NAT and Its Consequences

Since then, most home connections don't get their own public address at all. Internet providers place whole neighbourhoods behind a shared setup called **carrier-grade NAT (CGNAT)**, so hundreds or thousands of households quietly share a single public IPv4 address between them.

That has a real effect on DDoS defence: a security system that simply blocks "the IP address the attack is coming from" can end up blocking an entire apartment block's worth of innocent people who happen to share that address with an infected device.

It Cuts Both Ways

A single public IP hides which specific gadget behind it is infected, so defenders can't map a botnet one device at a time just by watching IPs. Attackers gain a layer of obscurity; defenders lose precision.

Why Anycast Is Unusual

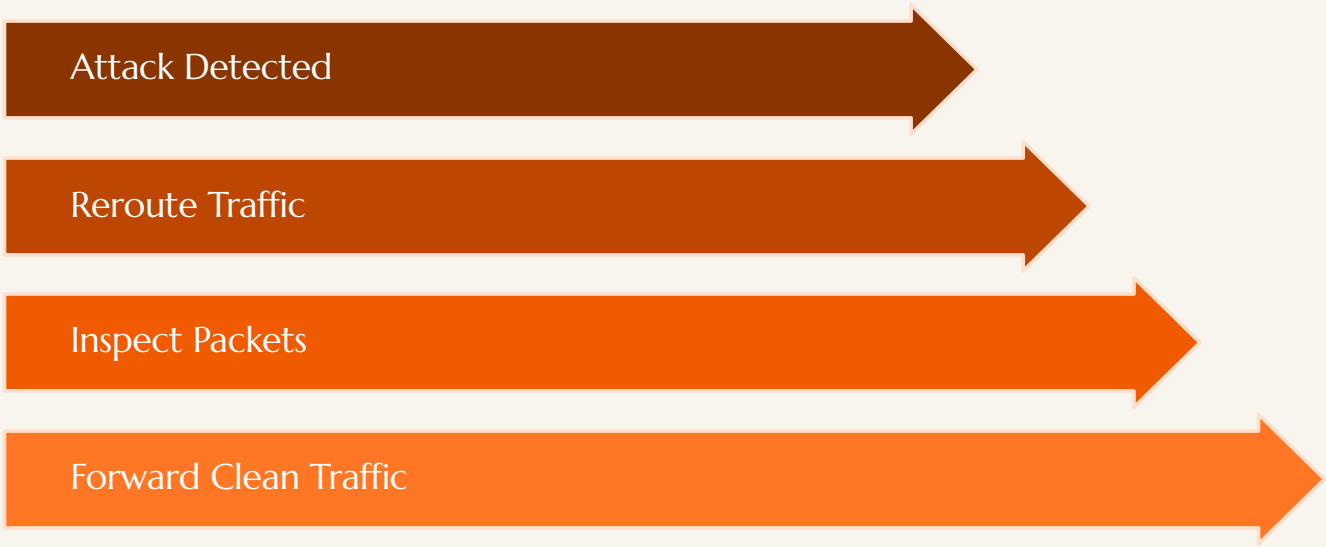
Cloudflare's trick of announcing one IP address from over **337 cities** at once is unusual rather than routine. Normally an IP address points to exactly one place; making it point to hundreds simultaneously isn't about having addresses to spare — it's about bending how internet routing works.

IPv4 exhaustion isn't just a technical footnote — it shapes how attacks are launched, how defenses are built, and why certain mitigation strategies are architecturally complex.

Scrubbing Data Centres and Mitigation

The Core Defense Architecture

Large companies handle volumetric attacks with a **scrubbing data centre** — a facility built to take dirty traffic in and send only clean traffic back out. The process itself has a name: **mitigation**, meaning everything a defender does to detect an attack, absorb it, and keep the real service running through it.



How Rerouting Works

When automated systems notice a spike, the network reroutes everything heading for the targeted website into a scrubbing data centre instead, so the flood never reaches the real server. Hardware there checks every packet at extremely high speed, deciding almost instantly whether it's a real visitor or junk — and only the genuine traffic gets sent on.

Specialised Hardware

Some of that hardware does just one job, fast, and nothing else. Other pieces can be reprogrammed on the fly, letting engineers respond to new attack patterns as mitigation is happening, without switching anything off. This adaptability is critical: attackers change tactics mid-attack, and static rule sets can't keep up.

Not All "DDoS Protection" Actually Filters Anything

What Null-Routing Actually Means

Real mitigation — the scrubbing kind — is expensive to build. Plenty of budget hosting providers advertise "DDoS protection" without doing anything like it. Their actual response to an attack is usually **null-routing**, also called **blackholing**.

The moment an attack is detected, the provider's upstream router is told to simply drop every packet addressed to the target's IP — attacker and genuine visitor alike — before any of it can reach the shared infrastructure other customers depend on. No hardware inspects a single packet, and nothing gets sorted into "real" and "junk." The site just goes dark for as long as the attack lasts.

Two Definitions of "Protected"

That isn't necessarily dishonest, so much as a difference in what "protected" means. From the hosting provider's side, the goal is met: the attack traffic no longer threatens other customers sharing the same network. From the site owner's side, the outcome looks identical to having no protection at all — since the attacker's real goal, taking the site offline, has just been handed to them for free.

How to Tell the Difference

The giveaway is usually in the fine print. Genuine scrubbing providers advertise a specific mitigation capacity in **Gbps or Tbps** and describe filtering. Budget providers advertising "DDoS protection" as a bare checkbox feature, with no numbers attached, are very often describing null-routing instead.

> ⚠ If a provider lists no capacity figures alongside their DDoS protection claim, assume null-routing until proven otherwise.

Getting a Feel for the Numbers

Network speed is measured in bits per second, and it climbs in jumps of a thousand. Understanding this ladder is essential for interpreting every attack size and mitigation capacity figure in this field.

1

Kbps — Kilobits per second

Old dial-up modems ran here, around **56 Kbps**. The baseline of consumer internet history. Essentially irrelevant to modern DDoS discussion except as a reference point for scale.

2

Mbps — Megabits per second

A decent home broadband connection — enough for a couple of HD video streams — sits somewhere around **25 to 100 Mbps**. This is the bandwidth of the devices that make up most botnets.

3

Gbps — Gigabits per second

A thousand times faster than Mbps. This is the territory of **business fibre lines** and the links inside a data centre. Small-to-medium DDoS attacks are measured here.

4

Tbps — Terabits per second

A thousand times faster than Gbps — so a **million times** a typical home connection. One Tbps is roughly enough bandwidth to stream well over a hundred thousand HD videos at the same instant. The largest attacks in history are measured here.

From here on, terabit-per-second figures are written as **Tbps**, the way engineers actually write them. Keep that ladder in mind — from home connections up through Gbps and into Tbps — every attack size is being measured against it.

Cloudflare

OVHcloud

Not every provider builds this the same way. Cloudflare and OVHcloud — two of the biggest names in DDoS protection — show just how differently the same problem can be solved.

Cloudflare's 337+ Global Server Locations

Cloudflare: Hide the Target

Cloudflare runs its network from over **337 cities worldwide**, and every single one of those locations announces the exact same IP address at the same time — a trick called **anycast**. When a botnet attacks a Cloudflare-protected site, the attack doesn't hit one building; it lands on whichever Cloudflare location is nearest to each attacking device.

An attack from Brazil and an attack from Vietnam get split into two smaller problems handled in two different cities. Every one of those locations can scrub and mitigate traffic on the spot — no single central scrubbing data centre to overwhelm or knock out.

In India, Cloudflare has **13 locations**: Ahmedabad, Bangalore, Chandigarh, Chennai, Hyderabad, Kannur, Kanpur, Kochi, Kolkata, Mumbai, Nagpur, New Delhi, and Patna.

500+

Tbps total capacity

337+

Cities worldwide

Attackers never learn where the real server actually sits — only where Cloudflare's edge is.

OVHcloud: Take the Hit Head-On

OVHcloud takes the opposite approach. Its servers keep their real, public address, and instead of hiding behind a global network, OVHcloud surrounds its own data centres with dedicated scrubbing hardware it calls the **VAC** (short for vacuum).

Suspicious traffic gets rerouted into these VAC nodes, which use specialised processors to inspect and filter packets before anything reaches the actual server. This suits customers like game servers, where players connect directly and hiding the address behind a proxy isn't really practical.

17+

Tbps mitigation capacity

100

Tbps backbone capacity

Same goal, opposite methods: one keeps attackers guessing where to aim, the other simply builds enough capacity — in the right place — to take the hit head on.

Closer to Home: DDoS Defence in India

National Governance Layer

At the national level, the **Indian Computer Emergency Response Team (CERT-In)**, operating under the Ministry of Electronics and Information Technology, is the country's designated agency for handling cyber incidents — DDoS attacks included — under the IT Act of 2000. Since 2022, its directions have required companies and government bodies to report a qualifying incident within **six hours** of detecting it.

Sitting alongside CERT-In is the **National Critical Information Infrastructure Protection Centre (NCIIPC)**, set up in 2014 specifically to watch over systems whose failure would hurt the country badly: power grids, banking networks, telecom, and transport. Where CERT-In handles the broad flow of incidents, NCIIPC narrows in on the sectors that can't afford to go down at all.

Domestic Peering

The **National Internet Exchange of India (NIXI)** runs nodes across Mumbai, Chennai, Bengaluru, and Kolkata — keeping Indian-to-Indian traffic inside the country instead of routing it out and back.

Cloudflare's 13 Indian Nodes

Cloudflare's 337-city network currently counts **thirteen real nodes inside India**: Ahmedabad, Bangalore, Chandigarh, Chennai, Hyderabad, Kannur, Kanpur, Kochi, Kolkata, Mumbai, Nagpur, New Delhi, and Patna. An attack aimed at an Indian visitor gets caught and scrubbed at whichever of these sits nearest, rather than needing to travel out to Singapore or Europe first.

OVHcloud's Indian Presence

OVHcloud ran a smaller Point of Presence in Mumbai from 2020, then opened a full data centre there a few years later — VAC scrubbing hardware included.

Kerala's Own Infrastructure

Two of those thirteen nodes — **Kochi and Kannur** — sit inside Kerala itself, so a good deal of the state's traffic never has to leave it to reach a scrubbing point. Underneath that sits a layer that's Kerala's own: **K-FON**, the state's fibre network, runs its network operations centre out of Infopark, Kochi. The **Kerala State Data Centre**, which hosts the state government's own services, sits inside Technopark, Thiruvananthapuram.

Beyond Web Pages: Spectrum, Transit, and Why This All Costs So Much

The Gap Cloudflare Spectrum Fills

Cloudflare's anycast trick works naturally for web traffic because a reverse proxy can sit in front of a site, decrypt and inspect HTTPS, and forward on only what looks legitimate. But a game server doesn't speak HTTP; it uses its own raw TCP or UDP connections. Cloudflare's answer is a separate product called **Spectrum** — it extends the same anycast network and scrubbing capacity to any TCP or UDP application, proxying whatever port a game server, VOIP system, or custom protocol needs. In effect, Spectrum lets a customer get OVH-style direct-connect performance while still keeping the real server address hidden, though the extra proxying hop adds a sliver of latency that some game studios would rather avoid.

The Real Cost: Geography, Not Just Hardware

To answer requests near their source and intercept attack traffic before it can travel far, a network like Cloudflare's needs a physical presence in hundreds of cities — which means either **peering** (direct interconnection agreements, often free or low-cost, but each one has to be negotiated and maintained) or **transit** (paying larger backbone carriers by the gigabit to move traffic across distances where no direct peering exists).

Every one of those 337+ locations also needs its own scrubbing hardware sitting mostly idle, since attacks are bursty and unpredictable, plus the staff and monitoring to keep it all ready around the clock. You're paying for hundreds of Tbps of spare capacity, sitting mostly idle across a planet's worth of peering and transit contracts, that might only get used for a few minutes at a time.

The Competitive Landscape

Fastly

Runs its own global anycast edge network — generally smaller in footprint than Cloudflare's but built around the same idea, with DDoS protection bundled alongside CDN and edge-computing services.

Akamai

One of the oldest content delivery networks, operates one of the largest distributed edge footprints on the internet with similarly high-capacity scrubbing.

AWS Shield & Azure DDoS Protection

Folded directly into their respective cloud platforms — defend resources hosted on AWS or Azure particularly well, but aren't standalone services for sites hosted elsewhere.

Imperva & Radware

Specialist vendors often chosen by enterprises that want DDoS mitigation as one part of a wider security package.

Meaningful DDoS protection is a scale business — that's why the market has stayed concentrated among a small number of large players.

Attacks That Don't Need to Be Big

Not every attack floods a website with sheer volume; some are quieter, and just as effective.

Application-Layer Attacks

A botnet can send only a small trickle of traffic, but if every request asks the search feature a question, something else breaks. Answering one question is easy for a database; answering ten thousand a second is not, and the processor gets pushed to its limit until the site slows or stops — not from the amount of data, but from the **work each request forces on the server**.

These attacks are harder to detect precisely because the traffic volume looks normal. Standard bandwidth-based thresholds never trigger. The attack succeeds through computational exhaustion rather than pipe saturation.

Two Tools That Catch Quiet Attacks

Web Application Firewall (WAF)

Reads what each request is actually *asking for*, rather than just how much traffic is arriving. Recognises known tricks attackers use to fool a database into handing over information it shouldn't — SQL injection patterns, malformed query strings, known exploit signatures.

Rate Limiting & CAPTCHA

Watches how fast each visitor is making requests. Ask faster than any human reasonably could, and it shows you a puzzle a real person solves without thinking and a bot usually can't. Separates human visitors from automated traffic at the behavioral layer.

The full picture of DDoS defense spans three layers: volumetric capacity (Tbps of scrubbing), network-layer filtering (firewalls, null-routing vs. real mitigation), and application-layer inspection (WAF, rate limiting). No single tool covers all three.

The Biggest Attacks on Record

DDoS attacks have grown at a startling pace.

October 2016

The **Mirai botnet**, built from hijacked security cameras and home routers, hit the DNS provider **Dyn** so hard that Twitter, Netflix, Reddit, and PayPal all went dark for hours, even though none of those sites were directly attacked. Dyn simply couldn't answer the address lookups needed to reach them. That same botnet hit **OVH** a month earlier at around **1 Tbps**, the biggest attack anyone had publicly seen at the time.

February 2018

GitHub was hit with **1.35 Tbps**, not from a botnet at all, but by abusing exposed **Memcached** servers, which can blow up a tiny request into a response tens of thousands of times bigger. GitHub's own defences rerouted the flood to **Akamai's Prolexic** mitigation service, and the site was back within about ten minutes.

May - September 2025

Cloudflare mitigated a **7.3 Tbps** attack in **May 2025**, then **11.5 Tbps** in September, then **22.2 Tbps** later that same month.

October 2025

Microsoft's Azure network absorbed a **15.72 Tbps** flood aimed at a single customer in Australia, the largest attack ever seen inside a cloud platform.

December 2025

A botnet named **Aisuru** (normally built from hundreds of thousands of hijacked routers, cameras, and other home devices with weak passwords, though this particular attack drew mainly on hijacked Android TV boxes) launched a **31.4 Tbps** attack at telecom companies, the largest ever recorded, delivering roughly **200 million requests** every second at its peak. Cloudflare mitigated it in 35 seconds, without a single engineer being paged. Set against the Kbps-to-Tbps ladder from earlier, that single attack carried more bandwidth than well over a hundred thousand typical home connections combined. None of these caused lasting outages, because the targets were sitting behind exactly the kind of scrubbing infrastructure described above; a decade earlier, an attack a fraction of that size took down some of the biggest names on the internet at once.

Why You Should Never Run a Public Server From Home

Suppose someone runs a small public website or game server from their house. A home connection usually carries **100 Mbps to 1 Gbps**; a botnet can throw far more than that without effort. The excess clogs the shared connection point where a whole neighbourhood's lines meet, so nearby neighbours may see slowdowns too, without knowing why. The home router never even sees most of the packets, because the provider's own protection usually blocks all traffic to that address further upstream, taking the entire home connection down until the attack ends.

> ⚠ Home connections cannot withstand botnet attacks, and provider-side protection can block the entire home connection.

Home plans aren't built with scrubbing data centres or any real mitigation, so anything meant to be public should sit behind a proper protection service instead.

Encryption and Randomness

Keeping a website standing is only half the story, though. Every packet still travels through networks that don't belong to you, its destination visible to anyone along the way. Staying online is one fight; keeping what you send unreadable as it crosses the world is another. That's where **encryption** comes in, and encryption only works if its keys are genuinely unpredictable, which depends on something computers are surprisingly bad at: being truly random.

In the next section, we'll look at how engineers generate that randomness, using a tool as unexpected as a wall of lava lamps.